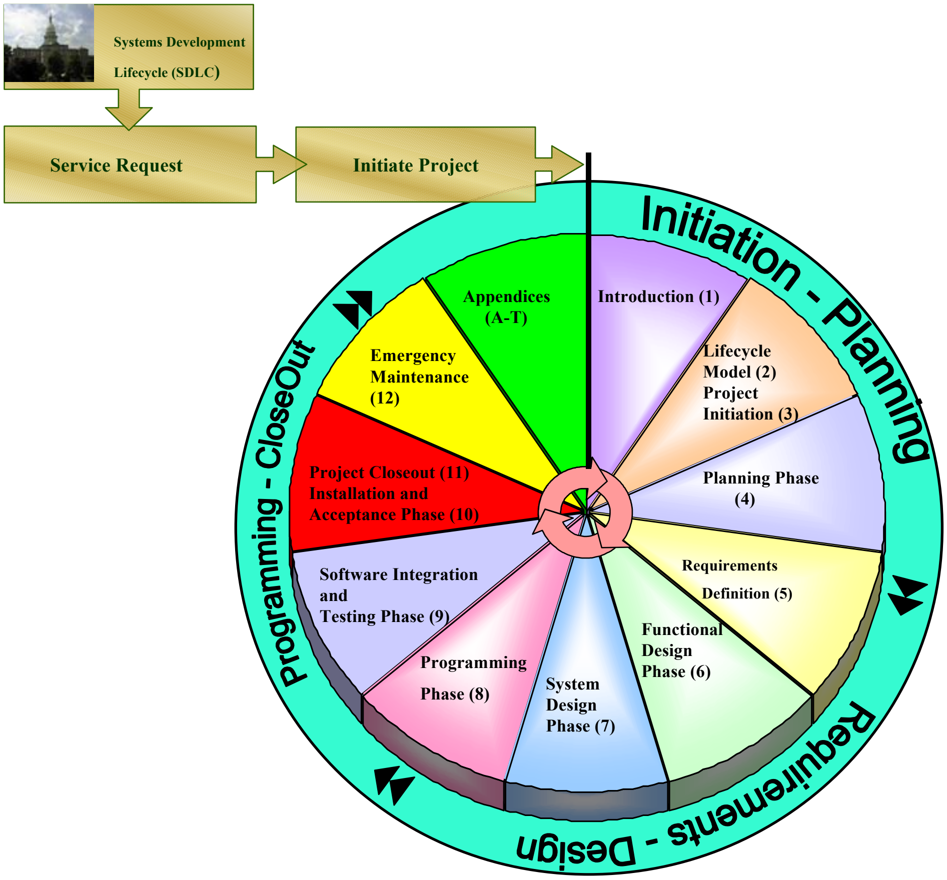
STATE OF MICHIGAN
SYSTEMS DEVELOPMENT LIFECYCLE
DESK REFERENCE


Systems Development
Lifecycle (SDLC)
Service Request
Initiate Project
Initiation - Planning
Requirements - Design
Programming - CloseOut
Introduction (1)
Lifecycle Model (2) Project Initiation (3)
Planning Phase (4)
Requirements Definition (5)
Functional Design Phase (6)
System Design Phase (7)
Programming Phase (8)
Software Integration and Testing Phase (9)
Project Closeout (11) Installation and Acceptance Phase (10)
Emergency Maintenance (12)
Appendices (A-T)

OFFICE OF RESEARCH AND POLICY

http://www.michigan.gov/dit

DECEMBER 2001

THIS DESK REFERENCE BELONGS TO:

NAME:____

AGENCY:____

PROJECT NAME:___________________________________

PROJECT MANAGER:________________________________

START DATE:________________________

FINISH DATE:________________________

SYSTEMS DEVELOPMENT LIFECYCLE DESK REFERENCE

PURPOSE

This guide was created to provide an easily transportable and customer friendly systems development lifecycle (SDLC) reference for agencies within the State of Michigan. The guide is generic enough to be applied to all development projects within the state. It is transferable from project to project but is not intended to be the sole source of information on Systems Development. For greater detail on any subject within this guide, refer to the State of Michigan Systems Development Lifecycle and the Systems Development Lifecycle.

WHY THE DESK REFERENCE SHOULD BE USED

Systems Development is a process that, like anything else, improves with practice and repetition. This guide is a basis for a standard suite of processes and associated documents that will facilitate the implementation and control of systems development phases at all levels of state government.

UPDATE PROCESS AND CYCLE OF THIS GUIDE

This edition of the Systems Development Lifecycle Desk Reference (December 2001) is its first release. This guide is updated on a regular basis as requested by the State's Systems Development Lifecycle Advisory Group (made up of agency representatives). The Lifecycle Advisory Group meets on a quarterly basis to discuss content changes.

Changes and improvements to this guide will be a product of input from the customer. If changes or additions need to be made, please contact your agency representative to the Lifecycle Advisory Group, or contact the DIT Office of Research and Policy, and discuss it with him or her. The representative will have the opportunity to take your idea to the next Lifecycle Advisory Group meeting and recommend the addition or change in the next version release.

AUDIENCE

The Systems Development Lifecycle Desk Reference is intended for top-level managers, experienced and non-experienced project managers, project mentors and coaches, project management instructors, project team members, technology-oriented project participants, project management offices, developers, analysts, vendors and any interested individual desiring to gain an overview insight into conducting system development activities and recording the necessary documentation for the project. This desk reference is considered an action guide that describes how to initiate a development project by stepping through the phases of a project and providing the necessary outlay of documents needed to support a particular development phase.

This Systems Development Lifecycle Desk Reference can be of significant benefit to improve one-on-one systems development training conducted by a project mentor or coach. It can also be used as a reference for developing train-the-mentor materials for the classroom.

POINTS OF CONTACT

Please forward any comments or questions to the Office of Research and Policy within the Department of Information Technology. The Office of Research and Policy can be reached at (517) 241-2926, or visit their web site at http://www.michigan.gov/dit.

Acknowledgements

The State of Michigan would like to acknowledge the following individuals and organizations that created this Systems Development Lifecycle. Without their input and hard work, this would not have been achieved.

INDIVIDUALS

Penny Dewey DMB, Office of Research and Policy	**David Hill** DMB, Office of Information Technology Services Division
Larry Bailey Department of Environmental Quality	**Sheila Kelley** Department of State
Vaughn Bennett DMB, Office of Project Management	**Christine Knapp** MAIN
Dan Buonodono DMB, Office of Project Management	**Linda Myers** Department of Community Health
Juan Chapa Department of Treasury	**Cathy Pelham** DMB, Office of Information Technology Services Division
Susan Felkowski MAIN	**Lucy Pline** MAIN
John Forslin Department of Natural Resources	**Ginger Schoettinger** Department of State

HOW TO USE THIS DESK REFERENCE

Application of this Guide

This desk reference provides an easy to use variety of checklists, visual cues, design features and clear language that will assist project managers and project teams throughout Michigan State Government.

This guide is a high-level representation of the State of Michigan Systems Development Lifecycle (SDLC). When more detail is needed, the customer should refer to the State of Michigan Systems Development Lifecycle, either in hard copy or electronically at or the http://www.michigan.gov/dit Intranet site.

Tips and Techniques for Use

This desk reference has been broken down into seven major sections that are tabularized along the right edge of the guide. The sections are itemized in the Table of Contents on the following pages.

The Overview Section provides high-level Systems Development Lifecycle materials and visual aids. Later in the guide, each phase, and the respective processes within that phase, is covered in greater detail.

🗐 Look for this symbol when reading the phase areas of the lifecycle. Whenever this symbol is found next to a *process* or *step* it means that there are deliverables associated with this step.

⌘ Look for this symbol when reading the phase areas of the lifecycle. Whenever this symbol is found next to a *process,* it means that the action performed is simply a process and there are no deliverables.

Adapt the sample forms and templates in this booklet as needed. They are simply a reference to more detailed forms and templates within the Systems Development Lifecycle. See the full version of the lifecycle for complete forms and templates.

– Special Project Notice –

Many business areas assume that a technology-based approach is the key to their business solution. Many project failures can be attributed to this assumption.

It is recommended that an “assessment” of the business environment take place prior to adopting a technical solution. This will ensure that the business area is ready to implement a technology-based solution.

Also, one must fully understand all project components prior to "executing the plan" so that nothing "falls through the cracks" during the Project Planning and Execution Phases.

Table of Contents

REQUIREMENTS DEFINITION PHASE

FUNCTIONAL DESIGN PHASE

SYSTEMS DESIGN PHASE

PROGRAMMING PHASE

SOFTWARE INTEGRATION AND TESTING PHASE

INSTALLATION AND ACCEPTANCE PHASE

Project Screening and Selection

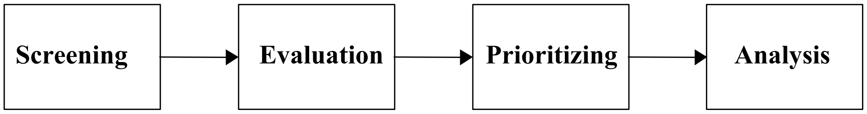
The Project Assessment Process

Every project starts with an idea. That idea may be the result of a unique thought or design, it may respond to a regulatory mandate, it may answer a call for operational maintenance, or it may be as simple as providing scheduled updates. In essence, projects are generated for many different reasons; however, projects warrant special consideration for uniqueness, importance, cost, priority, and duration of effort. Accordingly, potential projects, so as not to under estimate their 'value-add' and timing, need to be subjected to an assessment process that will allow the sponsor, stakeholders, project team, and other interested parties to validate the potential project benefits and timing.

Because many teams are initiated without regard for need and feasibility, an assessment process that includes valuation criteria should be pursued in order to ascertain the merit of the project itself. Major component phases of the assessment process can include, but may not be limited to, the items noted in Figure 1:

Figure 1

Project Assessment Phases



Screening

Typically, the screening phase consists of collecting data to determine whether or not the project belongs to a particular agency or organization and for preparing inputs for the Evaluation Phase. The perceived urgency of implementing ideas as a project will determine the timing delay in preparing data for review. This phase of the effort should be a quick and inexpensive exercise.

Evaluation

The Evaluation Phase builds on information gathered in the Screening Phase and provides, in greater detail, potential project information that will be used for evaluation. This information is then used to make such determinations as whether or not the idea warrants a project effort, integrates into the agency strategy, and fits within current budget constraints, and/or conflicts with ongoing projects. It will help detail the "protracted" benefits of the project. This phase may require the inputs of "outside" experts, the utilization of computational analysis, or it may include the use of technological forecasting. The results of the Evaluation Phase may indicate that the idea has reached an acceptable level to be considered a project. This would lead to the next step of prioritizing the implementation of this project with regard to the current agency workload.

Prioritizing

In the Prioritizing Phase, each idea (if there is more than one idea or if there is a comparison with ongoing projects) is weighted and appraised in terms of its relative strengths and weaknesses. This weighting would determine not only its individual merit as a project to pursue, but it would indicate a relative strength compared to ongoing or competing projects. In order to determine whether to pursue this project, a number of various techniques may be used. A few of the more generally accepted procedures are:

- Checklist/Scoring Models – a "spreadsheet" type analysis weighting various projects.
- Cost Benefit Analysis – a comparison of benefits from completing the project versus the outcomes of not instituting the project (this must be carefully considered when the benefits are difficult to measure; e.g., conducting a training seminar versus installing a "tele-file" system).
- Risk Analysis – an analysis of issues created while the potential project is being conceived. The intent of Risk Analysis is to try and quantify concerns that could possibly impede project progress and deter

outcome. (A most popular and useful technique used in analysis of a system is the Failure Modes and Effects Analysis–FMEA.)

- Decision Trees (flow networks) – a method for depicting and facilitating the analysis of problems that involves sequential decisions and variable outcomes over time.

It is hoped that any, or all of these, techniques will be useful in determining the relative merit of projects. Summarily, the results of this Prioritizing Phase will lead to an initial allocation of resources (human, capital, financial) toward beginning the efforts of the project.

Analysis

Analysis of enterprise considerations defines the final phase of project assessment selection. If the results of the Evaluation Phase indicate that the project should replace an ongoing project, then an analysis will need to be conducted as to how to reallocate resources to the new project while an ongoing project is temporarily put on hold or perhaps terminated. The process of going through an Analysis Phase will be used only if projects are determined that they will be competing for the same resources.

An Assessment Matrix

An Assessment Matrix, as referenced in the Prioritization Phase, provides a method for making decisions among alternatives based on their key components and benefits. When a senior executive must choose between two or more options, an assessment aid will provide straightforward, quantitative information which can be easily and quickly used to support decisions. Figure 2 displays an example of a completed weighting, assessment method that may be used in conjunction with agency generated criteria (see Figure 3 as an example) in determining relative merits of projects.

Figure 2

Project	Resources	Duration	Risk	Cost	Rating
Project New	**3**	**3**	**5**	**3**	**14**
Project 1	**1**	**1**	**1**	**3**	**6**
Project 2	**3**	**1**	**3**	**3**	**10**

Project 3	**5**	**3**	**3**	**3**	**14**
Project 4	**3**	**5**	**2***	**5**	**15**

*** Arbitrary decision**

Figure 3

Project Size	**Resources**	**Duration**	**Risk**	**Cost**
Small = 1	<5	< 3 months	No impact	< $50K
Medium = 3	< 10	< 6 months	Impacts Divisions	< $250K
Large = 5	> 10	> 6 months	Impacts other Agencies	> $250K

Ranking

A simple Likert ranking scale (1, 3, or 5) can be easily applied to choosing how projects are prioritized and implemented. The following ranking scale applies to the example above:

- A score of 4 – 8 = a small project
- A score of 9 – 15 = a medium project
- A score of 16 and higher = a large project

Because different Agencies have different internal requirements, it is suggested that each Agency determine the best lifecycle for implementing an assessment scheme for their use.

When Not to Formalize a Project Effort

The formalization of project efforts is as unique as there are numbers of projects being undertaken, and agencies undertaking them. However, it is generally accepted best practice that the establishment of project activities (scope, plan, WBS, scheduling, and other project components as described in this lifecycle) need not be formalized for efforts with less than three people whose duration does not exceed one month.

Essentially, it is recommended that an assessment approach be kept flexible enough so that the effort and results are consistent with the size and complexity of the alternatives being evaluated, life cycle phase, and level and type of review being supported.

Project Notebook and Checklist

PROJECT NOTEBOOK

The Project Notebook is the "Repository" that holds all project-related documentation. The Project Notebook is the central location for project-related information, and it contains current, up-to-date documentation, such as current project schedule, status reports, project change notices, etc. The Project Notebook also contains historical documentation such as base-lined versions of the project schedule, status reports, risk analyses, issues log, etc.

The Project Notebook can be in the form of a three-ring binder or a series of hierarchical folders on a secured network drive, or as part of a document management tool.

The Project Notebook should be developed as early as possible in the project to promote organization and documentation accountability. Contained in the Project Notebook is the Project Plan. The Project Plan comprises all of the documents, templates, and statements that are developed during the Planning Phase of the project. A format for the Project Plan is summarized as part of this Desk Reference. A more detailed version of this format may be found in the Systems Development Lifecycle document.

PROJECT NOTEBOOK CHECKLIST

A methodical and efficient project checklist can prevent the loss of time while trying to develop a plan for establishing the need for necessary project documentation. It is hoped that the Project Notebook Checklist, located on the following page, will prove useful as a place to begin.

TIPS AND TECHNIQUES FOR USE

The Project Notebook Checklist is broken down by the various phases of a project. The checklist is meant to facilitate the identification of the various documents necessary to be completed by the team during the course of the project. This checklist should be used as an aid to record and monitor progress of major requirements needed to document project activities.

The Project Notebook Checklist should serve as the Table of Contents for the Project Notebook. Miscellaneous Phase items can be placed immediately after the master section tabs (i.e., "Planning" in the case of the Project Plan Format Document). The contents of this guide should be filed with the project documentation during archiving (during the Project Closeout Phase).

Project Notebook Checklist

PHASE	PROJECT DOCUMENT	ASSIGN DATE	RESPONSIBLE PARTY	COMPLETE DATE
PROJECT INITIATION	PROJECT FEASIBILITY			
	PROJECT CONCEPT			
	PROJECT CHARTER			
PLANNING				
	PROJECT OBJECTIVES			
	PROJECT SCOPE			
	HIGH-LEVEL REQUIREMENTS			
	PROJECT PROFILE			
	FUNCTIONAL AREA CONTACTS			
	SUMMARY OF PLATFORM OPTIONS			
	FINAL RECOMMENDATIONS			
	SOFTWARE CONFIGURATION MANAGEMENT PLAN			
	SOFTWARE QUALITY ASSURANCE PLAN			
	CRITICAL SUCCESS FACTORS			
	WORK BREAKDOWN STRUCTURE			
	ORGANIZATIONAL BREAKDOWN STRUCTURE			
	COST BENEFIT ANALYSIS			
	RESOURCE PLAN			
	PROJECT SCHEDULE			
	RISK PLAN			
	PROCUREMENT PLAN			
	COMMUNICATIONS PLAN			
	PROJECT BUDGET ESTIMATE			
	PROJECT PLANNING TRANSITION CHECKLIST			
	TECHNICAL PROJECT COMPONENTS			

PHASE	PROJECT DOCUMENT	ASSIGN DATE	RESPONSIBLE PARTY	COMPLETE DATE
REQUIREMENTS DEFINITION	REQUIREMENT TRACEABILITY MATRIX			
	SOFTWARE CHANGE REQUEST FORMS			
	SOFTWARE CHANGE CONTROL LOG			
	ANALYSIS TECHNIQUE DESCRIPTION			
	FUNCTIONAL REQUIREMENTS			
	INPUT AND OUTPUT REQUIREMENTS			
	PERFORMANCE REQUIREMENTS			
	INTERFACE REQUIREMENTS			
	SYSTEM INTERFACE REQUIREMENTS			
	COMMUNICATIONS REQUIREMENTS			
	SECURITY AND ACCESS REQUIREMENTS			
	CONTINUITY OF OPERATIONS STATEMENT OR PLAN			
	DATA REQUIREMENTS			
	IMPLEMENTATION REQUIREMENTS			
	SOFTWARE REQUIREMENTS SPECIFICATION			
	PROJECT TEST PLAN			
	ACCEPTANCE TEST PLAN			
	DESIGN TECHNIQUE DESCRIPTION			
	PROJECT STATUS REPORTS			
	ISSUE DOCUMENT / ISSUES LOG			
	TECHNICAL PROJECT COMPONENTS			

PHASE	**PROJECT DOCUMENT**	**ASSIGN DATE**	**RESPONSIBLE PARTY**	**COMPLETE DATE**
FUNCTIONAL DESIGN	DESIGN ENTITY INFORMATION			
	DEPENDENCY INFORMATION FOR DESIGN ENTITIES			
	SYSTEM INPUT AND OUTPUT DESIGN			
	MENU HIERARCHY DESIGN			
	DATA ENTRY SCREEN DESIGNS			
	DISPLAY SCREEN DESIGN			
	ON LINE HELP DESIGN			
	SYSTEM MESSAGE DESIGN			
	SYSTEM INTERFACE DESIGN			
	SYSTEM SECURITY CONTROL DESIGN			
	LOGICAL MODEL			
	DATA FLOW DIAGRAM			
	EXPANDED MATRIX			
	APPROVED FUNCTIONAL DESIGN			
	SOFTWARE & HARDWARE PROCUREMENT RECORDS			JUSTIFICATIONS, APPROVALS, INVOICES, PURCHASE ORDERS
	ACQUISITION PLANS			
	INSTALLATION PLANS			

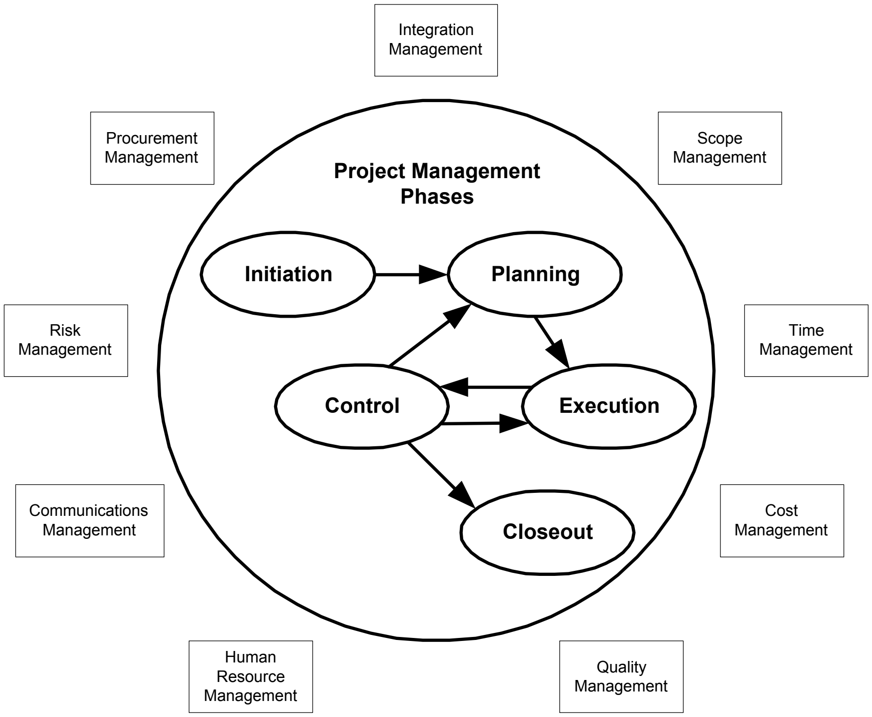
PHASE	PROJECT DOCUMENT	ASSIGN DATE	RESPONSIBLE PARTY	COMPLETE DATE
SYSTEM DESIGN	RECORDS FOR EACH ALTERNATIVE AND SUMMARY			
	ANALYSIS OF COSTS AND BENEFITS (ABC)			REPORT PROCESS USED AND ANALYSIS
	PRESENT THE PROJECT TEAM'S RECOMMENDATIONS FOR SYSTEM ARCHITECTURE			
	PHYSICAL MODEL			
	DATABASE STRUCTURE			
	EXPANDED MATRIX			
	APPROVED SYSTEMS DESIGN DOCUMENT			
	REVISED PROJECT PLAN			

PHASE	**PROJECT DOCUMENT**	**ASSIGN DATE**	**RESPONSIBLE PARTY**	**COMPLETE DATE**
Programming	Production Platform Acquisition Plan			
	Draft Installation Plan			
	Expanded Matrix			
	Approved Plans			
	Expanded Requirements Traceability Matrix			
	Draft Training Plan			
	Revised Project Plan			
Software Integration & Testing	Expanded Matrix			
	Final Integration Test Report			
	Approved Documents			Operations, Training, Installation
	System Test Materials			Inputs, Outputs, Results, Error Logs
	System Test Results			
	Revised Expanded Matrix			
	Approved Acceptance Test Plan			
	Pre-Acceptance checklist			
	Pre-Acceptance Security Issues			
	Draft Maintenance Plan			
	Revised Project Plan			

PHASE	PROJECT DOCUMENT	ASSIGN DATE	RESPONSIBLE PARTY	COMPLETE DATE
INSTALLATION AND ACCEPTANCE	INSTALLATION TEST MATERIALS			
	APPROVED TRAINING MATERIALS			
	TURN OVER ALL DOCUMENS TO MAINTENANCE STAFF			TRANSITION TO OPERATIONAL STATUS
	FINAL MAINTENANCE PLAN			
	STRUCTURED MANAGEMENT WALKTHROUGH REPORT			
	IPA REPORT			OPEN ISSUES
PROJECT CLOSEOUT	POST IMPLEMENTATION EVALUATION REPORT			
	TECHNICAL PROJECT COMPONENTS			
EMERGENCY MAINTENANCE	MODIFICATION REQUESTS			
	VALIDATED MODIFICATION REQUEST			
	UNIT & INTEGRATION TESTING PERFORMED			

Best Practice Notes

PROJECT MANAGEMENT PHASES AND KNOWLEDGE AREAS



WHAT IS PROJECT MANAGEMENT?

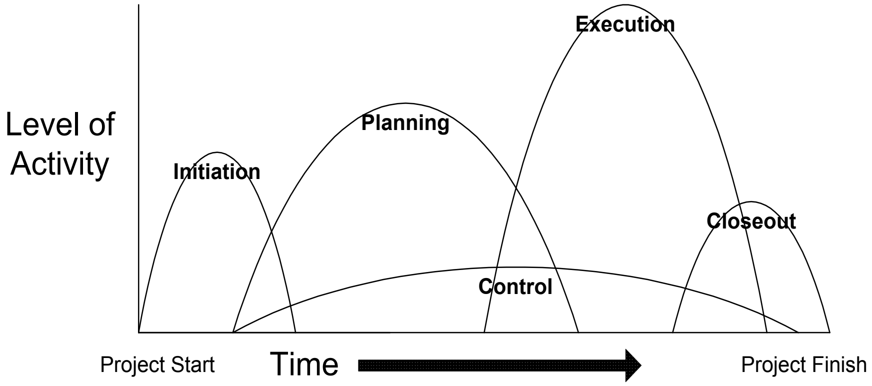
Project Management is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements.

PMBOK®, 2000

Characteristics of the Project Life Cycle

All projects are unique. As such, each project takes on a different form and presents many degrees of uncertainty. Therefore, managing these projects requires that organizations usually divide these projects into more manageable pieces called "phases". These phases allow the project team to provide better management and control in order to provide efficient and productive efforts throughout the life of the project. Collectively, these phases are sometimes called the "project life cycle".

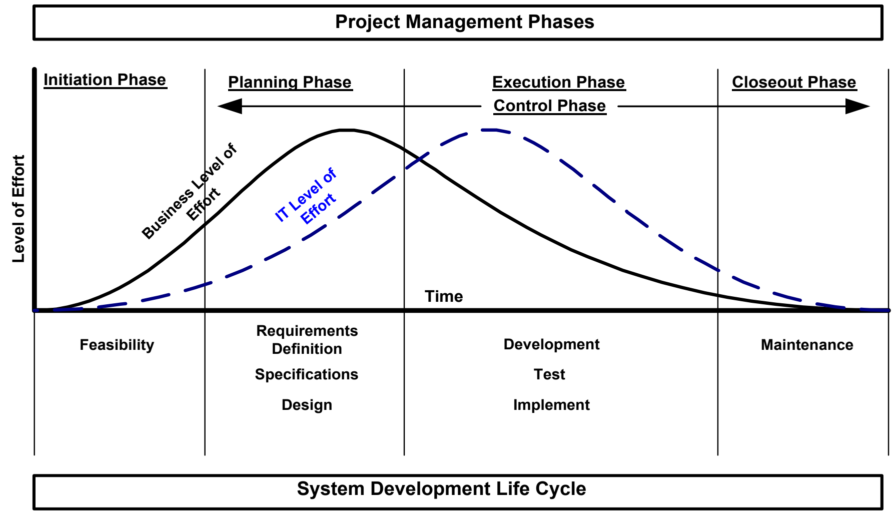
Although these project phases have been established to complement the project teams' involvement with the tasks, these phases are not stand alone as may be indicated in the "phases" portion depicted in the figure on page 8. These phases overlap and can typically resemble something as portrayed in the following figure.



Project Management and The System Development Life Cycle

The Systems Development Lifecycle that is summarized within this guide can be used on any type of development project. However, with the immersion of technology into almost every aspect of the way the State of Michigan does business, Information Technology (IT) Projects have become the most prevalent types of new project efforts. These projects include software development, telecommunications, and hardware installation.

Information Technology projects typically have their own life cycle called the *System Development Life Cycle (SDLC)*. The System Development Life Cycle has its own types of deliverables and processes that are focused on the more technical portions of the project. The diagram below is a graphical representation of the Project Management Phases (top portion) as compared to the System Development Life Cycle (bottom portion). Note that the two processes are performed in parallel and not in place of one another.



Project Management is a combination of processes to aid the project manager in order to guarantee complete and correct project management, while the technical staff typically carries out the System Development Life Cycle. Because of the large number of Information Technology projects within the agencies, this guide makes several references to the processes that take place at the same time as Project Management efforts. These sections are clearly labeled throughout the desk reference for your use.

Project Stakeholders and Their Responsibilities

To ensure project success, the project management team must identify stakeholders early in the project, determine their needs and expectations, and manage those expectations over the course of the project. Below is a list of possible project stakeholders presented in order of relevance to the use of this document.

Stakeholders on every project include:

Project Manager – who has ultimate responsibility for ensuring project success in all areas.

Customer – person(s) or organization(s) who use the product/services of the project.

Project Team Members – who are responsible for performing the work on the project.

Project Sponsor – who leads in getting the need for the project recognized and providing financial resources.

State Agency Management – who defines the business needs of the project.

Configuration Management – who manages the deliverable changes within the boundaries of the project.

Quality Assurance – who verifies the ability of the product/process to meet the stated requirements.

Agency/State Procurement – who provides purchasing and contract support to the project.

State of Michigan – who is responsible for defining statewide project policies and providing project oversight.

State of Michigan Citizens and Visitors – who are interested in the success of all projects.

See the State of Michigan Systems Development Lifecycle for a detailed listing of Roles and Responsibilities.

Selection of a Project Manager

Selection of a project manager is not easy nor is it something that should be taken lightly. A project manager's skills and actions are a direct reflection of the agency's commitment and competence in project management. A project manager's daily responsibilities typically include some or all of the following:

- Providing direction, leadership, and support to project team members in a professional manner at project, functional, and task levels
- Using, developing, and improving upon the Systems Development Lifecycle within the agency
- Providing teams with advice and input on tasks throughout the project including documentation, creation of plans, schedules, and reports
- Resolving conflicts within the project between resources, schedules, etc.
- Influencing customers and team members in order to get buy-in on decisions that will lead to the success of agency projects
- Delegating responsibility to team members

Taking these responsibilities into account, it is easy to see that a project manager should not necessarily be selected from an agency based strictly upon tenure or function, but rather based upon a combination of other strengths. A project manager should be selected based upon:

- Skills in project management methods and tools
- Interpersonal and team leadership skills
- Basic business and management skills
- Experience within the project’s technical field
- Respect and recognition among peers within the agency

Selecting a project manager based upon these criteria alone would be difficult, therefore more thought needs to be put into the process. Additional information would include level of experience, visibility of the project, availability, and personal interest on the part of the candidate.

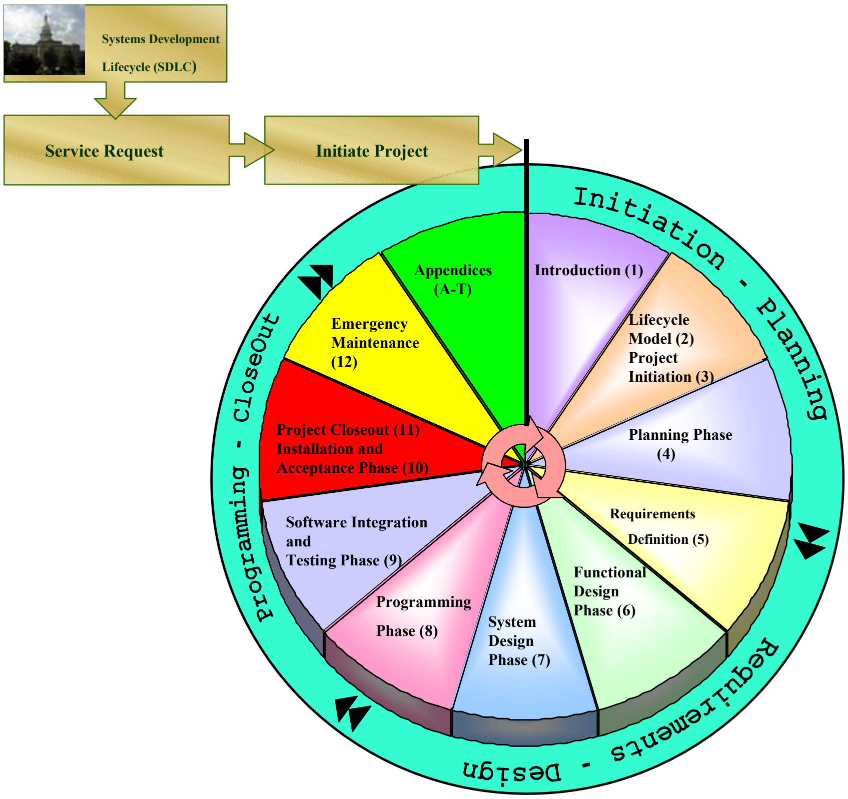
Project Phases and Deliverables

The diagram on the following page is a more detailed version of the process introduced on page 8. This diagram is intended to give the customer a visual understanding of the process and document flow through the project phases. Each box is representative of one of the five phases of the project management process. Within each box, the reader will find one to three items:

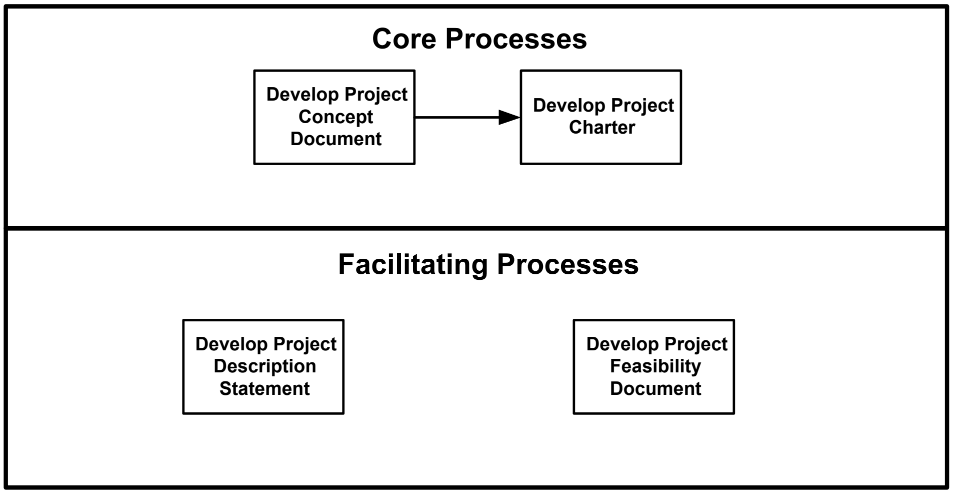
- A brief description of the phase, its relevance, and its purpose within the project.

- A list of the documents associated with that particular phase of the project. These are actual documents that will have to be created for each project, dependent on the needs or size of the project. Further descriptions of the documents are available later in the guide as each phase is broken down into greater detail.

- Information Technology Project Components – As previously mentioned, Information Technology projects are the most common projects within the State of Michigan. This section is dedicated to relating the deliverables and processes that are simultaneously carried out within the Systems Development Lifecycle.


Systems Development Lifecycle (SDLC)
Service Request
Initiate Project
Initiation - Planning
Requirements - Design
Programming - CloseOut
Introduction (1)
Lifecycle Model (2) Project Initiation (3)
Planning Phase (4)
Requirements Definition (5)
Functional Design Phase (6)
System Design Phase (7)
Programming Phase (8)
Software Integration and Testing Phase (9)
Project Closeout (11) Installation and Acceptance Phase (10)
Emergency Maintenance (12)
Appendices (A-T)

Project Initiation Phase



Core Processes
Develop Project Concept Document
Develop Project Charter
Facilitating Processes
Develop Project Description Statement
Develop Project Feasibility Document

🗐 = Deliverable ⌘ = Process

The Project Initiation Phase is the conceptual element of project management. The purpose of the Initiation Phase is to specify what the project should accomplish and to gain management approval.

PROJECT MANAGER ROLES AND RESPONSIBILITIES (OR LEAD PERSON IF PROJECT MANAGER NOT YET SELECTED)

- Develop Project Concept Document and Project Charter
- Define project success criteria
- Document project constraints
- Document project assumptions

INITIATION PHASE CORE PROCESSES

🗐 Develop the Project Concept Document – The Project Concept Document is the foundation for making the decision to initiate a project. It describes the project purpose and high level planning information to determine project viability.

🗐 Develop the Project Charter – The Project Charter is created to formally communicate the existence of the project. The Project Charter is issued at the end of the Initiation Phase / beginning of the Planning Phase of a project and is used as the basis to create the Project Plan.

INITIATION PHASE FACILITATING PROCESSES

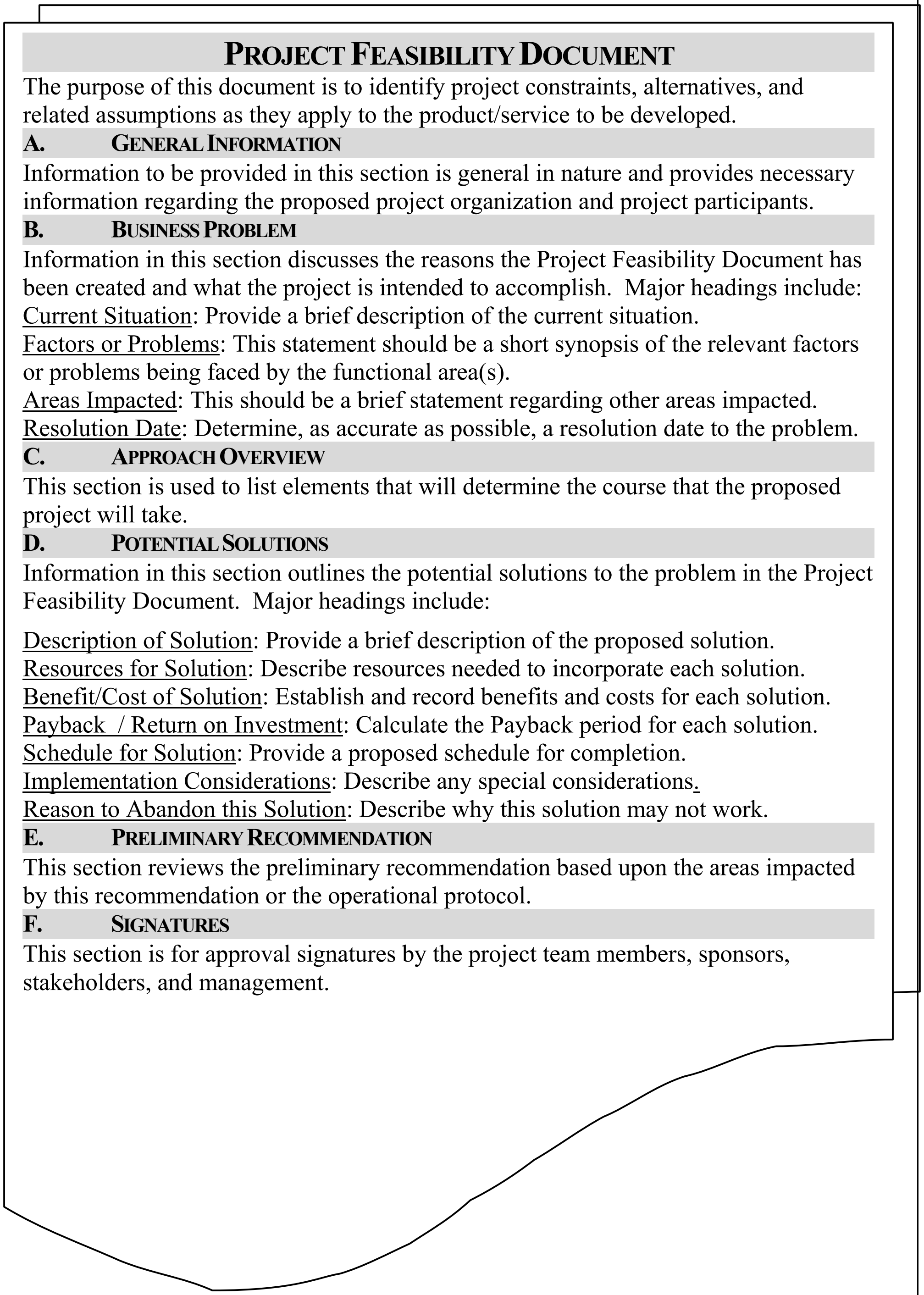
- Develop the Product Description Statement – The Product Description Statement is an informal, high-level statement that describes the characteristics of the product/process to be created. The Product Description Statement leads to the development of the Project Concept Document.
- Develop the Project Feasibility Document – The purpose of feasibility is to identify project constraints, alternatives, and related assumptions of the proposed product, to gain management approval, and to have management include the approval in the business plan.

INITIATION PHASE DELIVERABLES

- Project Feasibility Document – used in the beginning of the project cycle to analyze and discuss the feasibility of a project. The entire Project Feasibility Document Template is available in the State of Michigan Systems Development Lifecycle. See page 20 for an example of its contents.
- Project Concept Document – delineates a Product Description Statement, Critical Success Factors, and other top-level planning information. The entire Project Concept Document Template is available in the State of Michigan Systems Development Lifecycle. See page 21 for an example of its contents.
- Project Charter – used to formally initiate a project. The Project Charter Template is available in the State of Michigan Systems Development Lifecycle. See page 22 for an example of its contents.

INITIATION PHASE – OTHER IMPORTANT INFORMATION

- If the project is selected for implementation, the project manager should be selected when the idea becomes a project or at least by the end of the Initiation Phase.
- Be aware of barriers/problems that may arise in the Initiation Phase such as: Team Frustration, Obtaining Management Commitment, Customer Indecision, Scarcity of Resources, Lack of Coordinated Leadership, Lack of Consensus on Objectives, and Lack of a Management Sponsor.



PROJECT FEASIBILITY DOCUMENT

The purpose of this document is to identify project constraints, alternatives, and related assumptions as they apply to the product/service to be developed.

A. GENERAL INFORMATION

Information to be provided in this section is general in nature and provides necessary information regarding the proposed project organization and project participants.

B. BUSINESS PROBLEM

Information in this section discusses the reasons the Project Feasibility Document has been created and what the project is intended to accomplish. Major headings include:
Current Situation: Provide a brief description of the current situation.
Factors or Problems: This statement should be a short synopsis of the relevant factors or problems being faced by the functional area(s).
Areas Impacted: This should be a brief statement regarding other areas impacted.
Resolution Date: Determine, as accurate as possible, a resolution date to the problem.

C. APPROACH OVERVIEW

This section is used to list elements that will determine the course that the proposed project will take.

D. POTENTIAL SOLUTIONS

Information in this section outlines the potential solutions to the problem in the Project Feasibility Document. Major headings include:

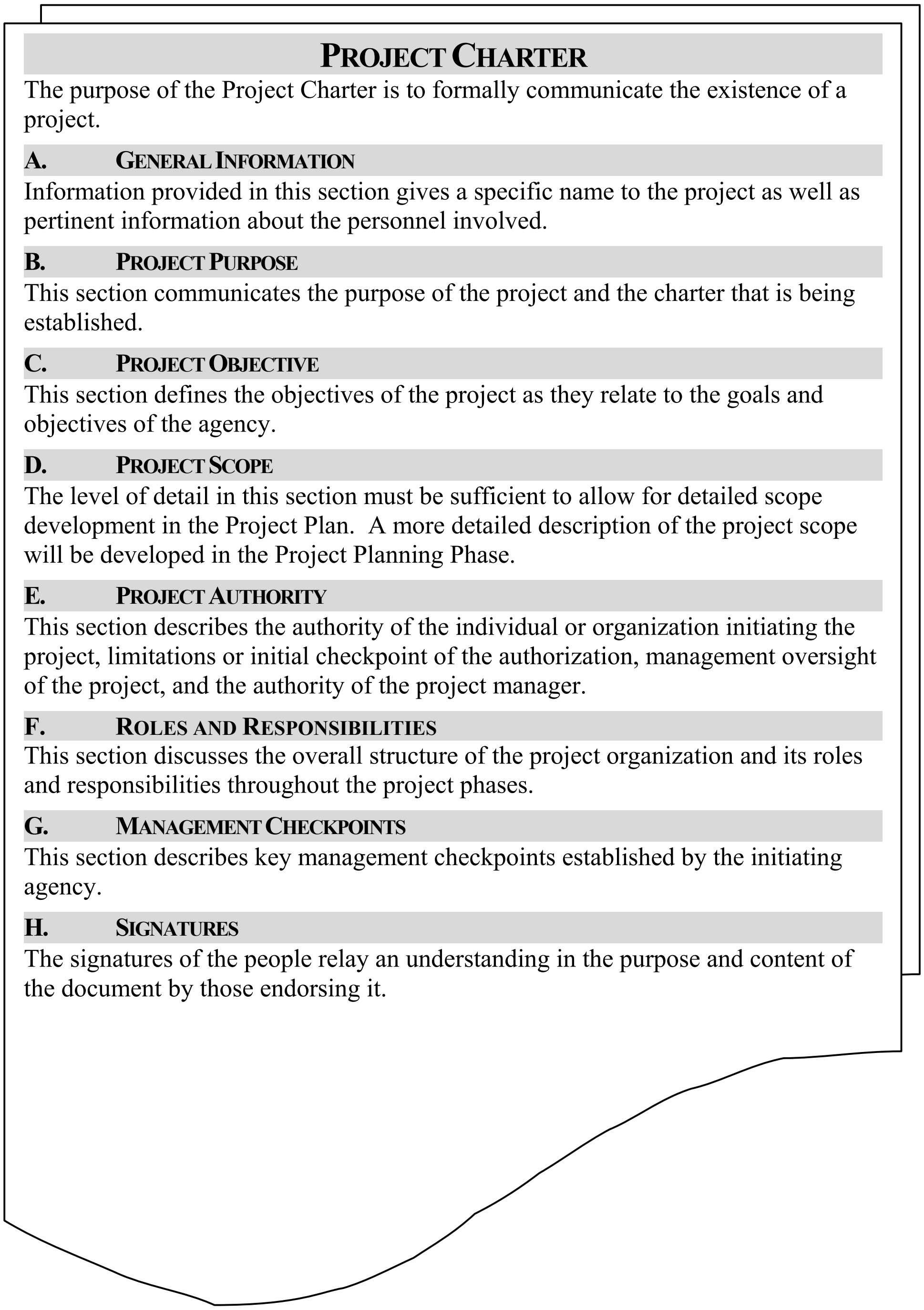
Description of Solution: Provide a brief description of the proposed solution.
Resources for Solution: Describe resources needed to incorporate each solution.
Benefit/Cost of Solution: Establish and record benefits and costs for each solution.
Payback / Return on Investment: Calculate the Payback period for each solution.
Schedule for Solution: Provide a proposed schedule for completion.
Implementation Considerations: Describe any special considerations.
Reason to Abandon this Solution: Describe why this solution may not work.

E. PRELIMINARY RECOMMENDATION

This section reviews the preliminary recommendation based upon the areas impacted by this recommendation or the operational protocol.

F. SIGNATURES

This section is for approval signatures by the project team members, sponsors, stakeholders, and management.



PROJECT CHARTER

The purpose of the Project Charter is to formally communicate the existence of a project.

A. GENERAL INFORMATION

Information provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. PROJECT PURPOSE

This section communicates the purpose of the project and the charter that is being established.

C. PROJECT OBJECTIVE

This section defines the objectives of the project as they relate to the goals and objectives of the agency.

D. PROJECT SCOPE

The level of detail in this section must be sufficient to allow for detailed scope development in the Project Plan. A more detailed description of the project scope will be developed in the Project Planning Phase.

E. PROJECT AUTHORITY

This section describes the authority of the individual or organization initiating the project, limitations or initial checkpoint of the authorization, management oversight of the project, and the authority of the project manager.

F. ROLES AND RESPONSIBILITIES

This section discusses the overall structure of the project organization and its roles and responsibilities throughout the project phases.

G. MANAGEMENT CHECKPOINTS

This section describes key management checkpoints established by the initiating agency.

H. SIGNATURES

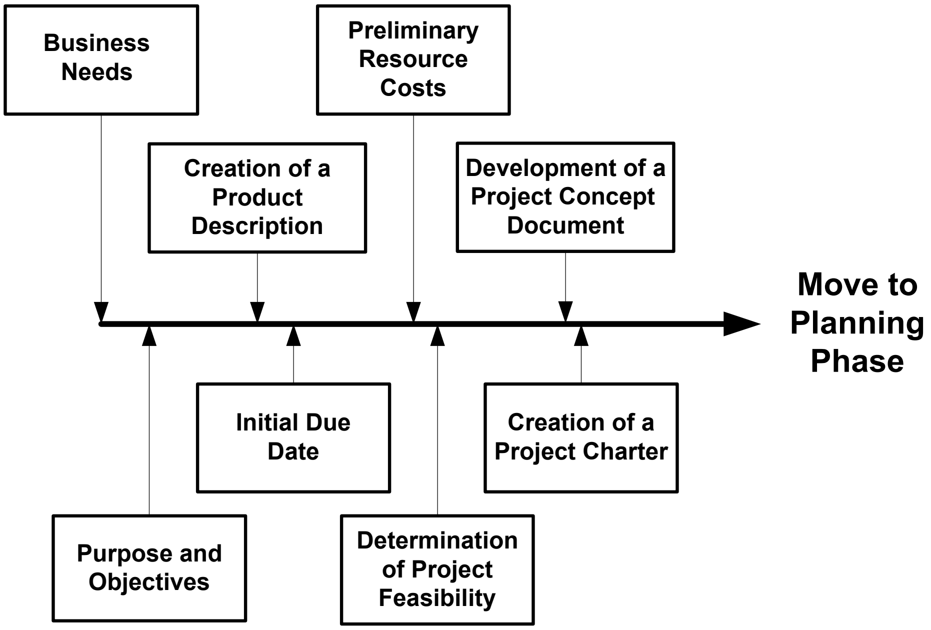
The signatures of the people relay an understanding in the purpose and content of the document by those endorsing it.

Considerations in the Timeframe for Completion

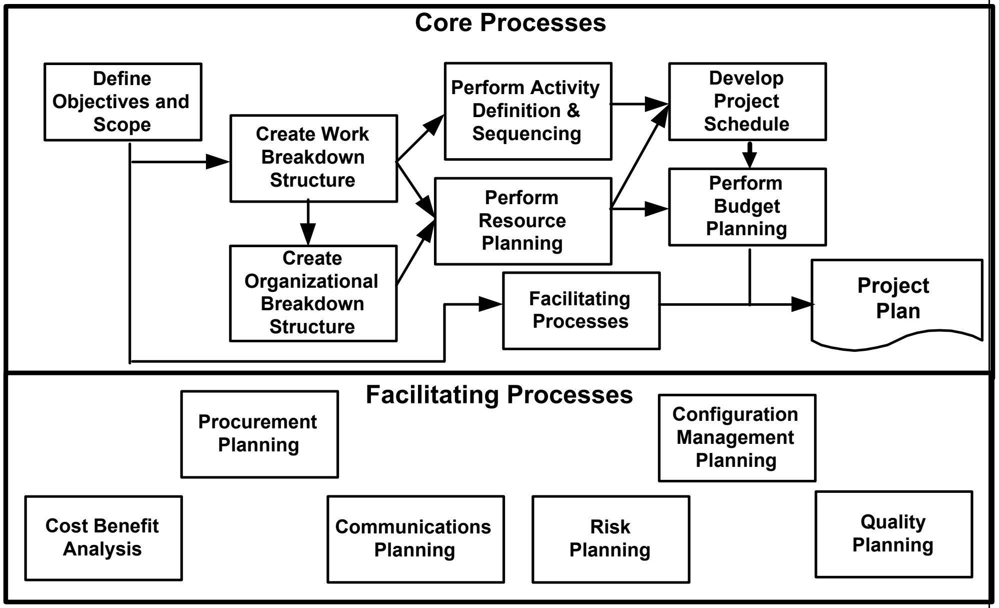
The general timeframe for most project teams to complete the process to generate a Project Charter varies widely according to project size, which is driven by unlimited factors. A generic sequence flow showing where basic processes of the initiation activities (through Project Charter to the Project Planning Phase) starts and ends is provided on the following diagram.

This sequence flow is provided only as a guideline since each project is unique and will require different levels of detail, research, and development. It does not take into consideration all levels of review that may vary between agencies.

This general sequence should be adjusted as necessary to reflect the specific project at hand.


Business Needs
Preliminary Resource Costs
Creation of a Product Description
Development of a Project Concept Document
Move to Planning Phase
Initial Due Date
Creation of a Project Charter
Purpose and Objectives
Determination of Project Feasibility

Project Planning Phase



🗐 = Deliverable ⌘ = Process

The purpose of the Project Planning Phase is to identify and document scope, business requirements, tasks, schedules, costs, risk, quality, and staffing needs. This process should continue until all relevant areas of the chartered project have been addressed.

PROJECT MANAGER ROLES AND RESPONSIBILITIES

- Develop detailed Project Plan, tailoring lifecycle to reflect project needs.
- Creation of a Work Breakdown Structure and Organizational Breakdown Structure.
- Develop or assist in the development of a Scope Management Plan, Project Schedule, Budget Estimate, Communications Plan, Risk Analysis, Configuration Management Plan Procurement Plan, Quality Plan, and Contingency Plan.
- Ensure that management, customers, affected state agencies and contractors agree to project commitments.
- Ensure that Project Plan is approved and baselined.
- Assign resources to project and assign work packages.
- Approve the Project Quality Plan and the Configuration Management Plan.

PLANNING PHASE CORE PROCESSES

- 🗐 Define Objectives and Scope – Define the need within the agency to understand the objective of the product/process being created and where this new project fits into the agency objectives. This is also the time when business requirements are defined.
- 🗐 Create the Work Breakdown Structure (WBS) – The Work Breakdown Structure is a deliverable-oriented document that is used to break down the work to be done within the project to a manageable level.
- 🗐 Create Organizational Breakdown Structure (OBS) - An Organizational Breakdown Structure represents the project organizational structure, arranged and coded in a hierarchical format to improve communication throughout the project. The Organizational Breakdown Structure assists in reporting project attributes that are the responsibility of an agency.
- ⌘ Activity Definition and Sequencing – Activity Definition and Sequencing involves dividing the project into smaller, more manageable components (activities) and then specifying the order of completion.
- ⌘ Perform Resource Planning – The resource component includes the ability to plan and manage the resources required to deliver the project. This starts with the agency selection and assignment of a project manager and includes the management of the resources assigned to that manager.
- 🗐 Develop Project Schedule – The Project Schedule provides a graphical representation of predicted tasks, milestones, dependencies, resource requirements, task duration, and deadlines. The project's master schedule interrelates all tasks on a common time scale.
- 🗐 Perform Budget Planning – Budget Planning is the determination of costs associated with the defined activities. The steps associated with budgeting are highly dependent upon both the estimated lengths of tasks and the resources assigned to the project. This process results in a documented budget estimate.

PLANNING PHASE FACILITATING PROCESSES

- ⌘ Cost Benefit Analysis – A Cost Benefit Analysis provides information to make a balanced decision about the cost and benefits, or value, of various economic choices about various alternatives within project activities/tasks.
- 🗐 Procurement Planning – Procurement Planning is the process in which the project manager identifies those needs of the project that can be met by purchasing products or services from outside their agency. Many agencies will have a procurement or contracts area, but on a large effort, it may still be necessary to outline the guidelines for procurement within the project.
- 🗐 Communication Planning – Communications Planning involves defining the information needs of project stakeholders as well as which people need what information, when it will be needed and how they will get it. A Communications Plan will be drafted as a result of this effort.

- Risk Planning – Risk Planning involves the following: risk identification, risk analysis and quantification, risk mitigation planning, and risk response. A Risk Management Worksheet is helpful in identifying and controlling these items.
- Configuration Management (CM) Planning – Configuration Management Planning is the process of managing changes to the product. Accordingly, it provides the project team with a change management lifecycle for identifying and controlling the functional and physical design characteristics of a deliverable. A Configuration Management Plan should be drafted as a result of this effort.
- Quality Planning – Quality Planning is the process of identifying which quality standards are relevant to the project and determines how to satisfy them. The result of this process is a document called the Quality Plan.

PLANNING PHASE DELIVERABLES

- Project Plan – A Project Plan is a formal, consolidation of Project Planning documents that is used to manage and control a project. It should be as accurate and complete as possible without being several volumes in length. The Project Plan documents the pertinent information associated with the project. A summarized version of the Project Plan, and its parts, is included on pages 28 through 30.
- Project Planning Transition Checklist – The Project Planning Transition Checklist ensures that planning activities have been finished, reviewed, and signed off so that the project may move into the Execution Phase. It should be organized according to the major areas of concern that will determine the project's success. See page 44.

SYSTEM DEVELOPMENT LIFE CYCLE COMPONENTS OF THE PLANNING PHASE

- Requirements Document – The Requirements Document is a formal document that outlines the high level requirements of a technical project.
- Specifications Document – The Specifications Document is a formal document that can provide specific information about the project deliverable characteristics. (Specifications might include measurements of time, weight, processing speed, etc.)
- Design Documents – Design Documents are technical documents that lay out in great detail the anticipated design of the project deliverable. (Examples might include technical schematics or construction plans.)
- Implementation Plan – At the completion of the Planning Phase, an Implementation Plan integrates the delineation of project tasks into an action plan for the project. A Training Plan outlines any required training to be completed by both the customer and members of the project team.

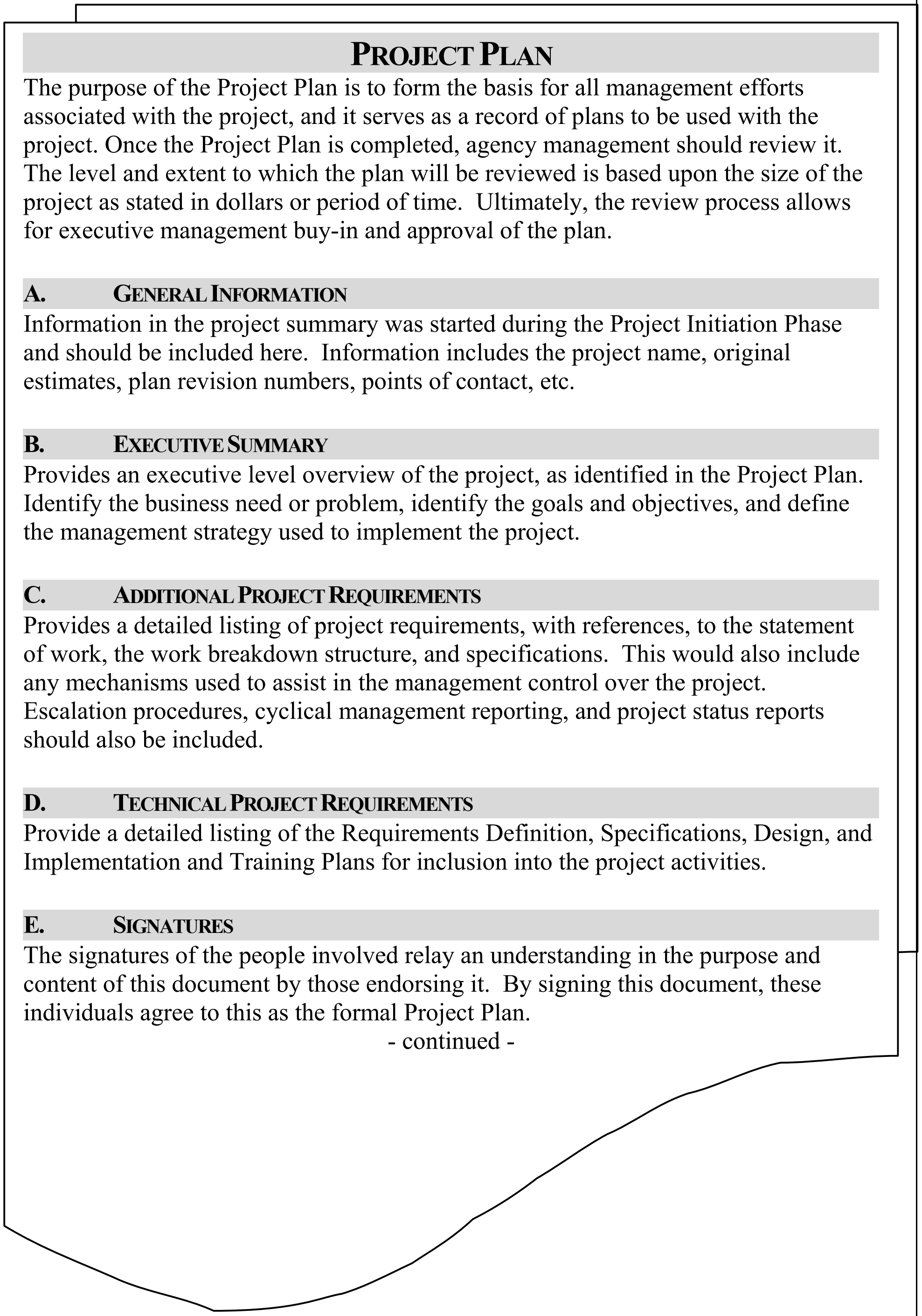
Planning Phase – Other Important Information

- ⌘ The Project Planning Phase is not something that should be done in a vacuum. Ensure buy-in by including all key stakeholders and project team members.
- 🗐 The Project Plan and other documents are not static documents and Project Planning is an iterative process. You should expect plans to be revised and improved upon consistently throughout the project. The Project Plan, once finished, should only be modified based on approved scope changes, as addressed in the Project Control Phase.
- ⌘ Keep the stakeholders informed about the progress of the Project Planning Phase. Often time people do not get interested in a project until it is in the Execution Phase and the importance of good planning is often overlooked.

Planning Phase – Other Important SDLC

The following tasks are performed in addition to the project management tasks in the planning phase:

Establish Communications with Functional Areas
Develop Project Plan
Develop Software Quality Assurance Plan
Develop Configuration Management Plan
Investigate Software Alternatives
Investigate Hardware Alternatives
Formulate Platform Options
Conduct Project Reviews



PROJECT PLAN

The purpose of the Project Plan is to form the basis for all management efforts associated with the project, and it serves as a record of plans to be used with the project. Once the Project Plan is completed, agency management should review it. The level and extent to which the plan will be reviewed is based upon the size of the project as stated in dollars or period of time. Ultimately, the review process allows for executive management buy-in and approval of the plan.

A. GENERAL INFORMATION

Information in the project summary was started during the Project Initiation Phase and should be included here. Information includes the project name, original estimates, plan revision numbers, points of contact, etc.

B. EXECUTIVE SUMMARY

Provides an executive level overview of the project, as identified in the Project Plan. Identify the business need or problem, identify the goals and objectives, and define the management strategy used to implement the project.

C. ADDITIONAL PROJECT REQUIREMENTS

Provides a detailed listing of project requirements, with references, to the statement of work, the work breakdown structure, and specifications. This would also include any mechanisms used to assist in the management control over the project. Escalation procedures, cyclical management reporting, and project status reports should also be included.

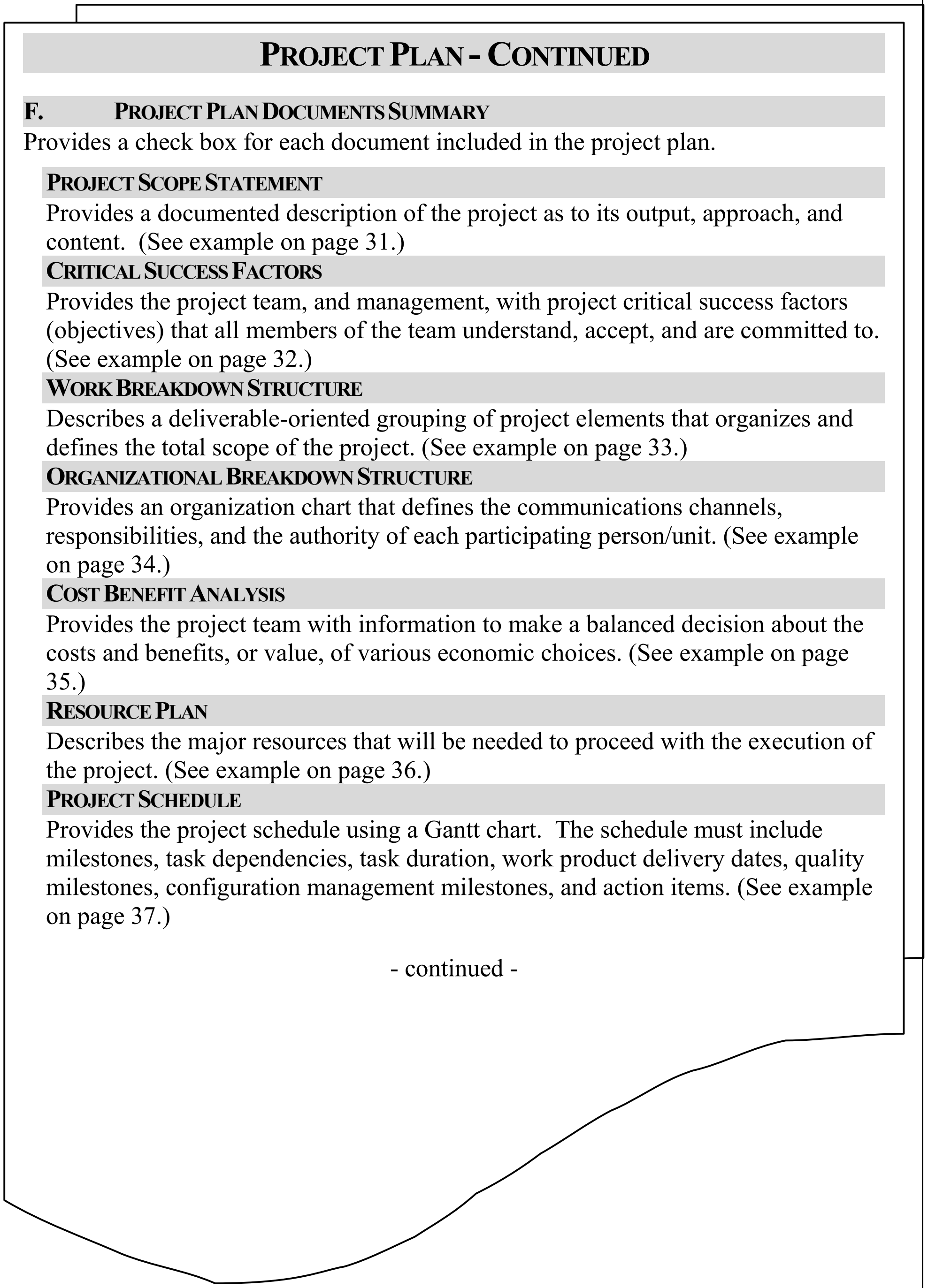
D. TECHNICAL PROJECT REQUIREMENTS

Provide a detailed listing of the Requirements Definition, Specifications, Design, and Implementation and Training Plans for inclusion into the project activities.

E. SIGNATURES

The signatures of the people involved relay an understanding in the purpose and content of this document by those endorsing it. By signing this document, these individuals agree to this as the formal Project Plan.

- continued -



PROJECT PLAN - CONTINUED

F. PROJECT PLAN DOCUMENTS SUMMARY

Provides a check box for each document included in the project plan.

PROJECT SCOPE STATEMENT

Provides a documented description of the project as to its output, approach, and content. (See example on page 31.)

CRITICAL SUCCESS FACTORS

Provides the project team, and management, with project critical success factors (objectives) that all members of the team understand, accept, and are committed to. (See example on page 32.)

WORK BREAKDOWN STRUCTURE

Describes a deliverable-oriented grouping of project elements that organizes and defines the total scope of the project. (See example on page 33.)

ORGANIZATIONAL BREAKDOWN STRUCTURE

Provides an organization chart that defines the communications channels, responsibilities, and the authority of each participating person/unit. (See example on page 34.)

COST BENEFIT ANALYSIS

Provides the project team with information to make a balanced decision about the costs and benefits, or value, of various economic choices. (See example on page 35.)

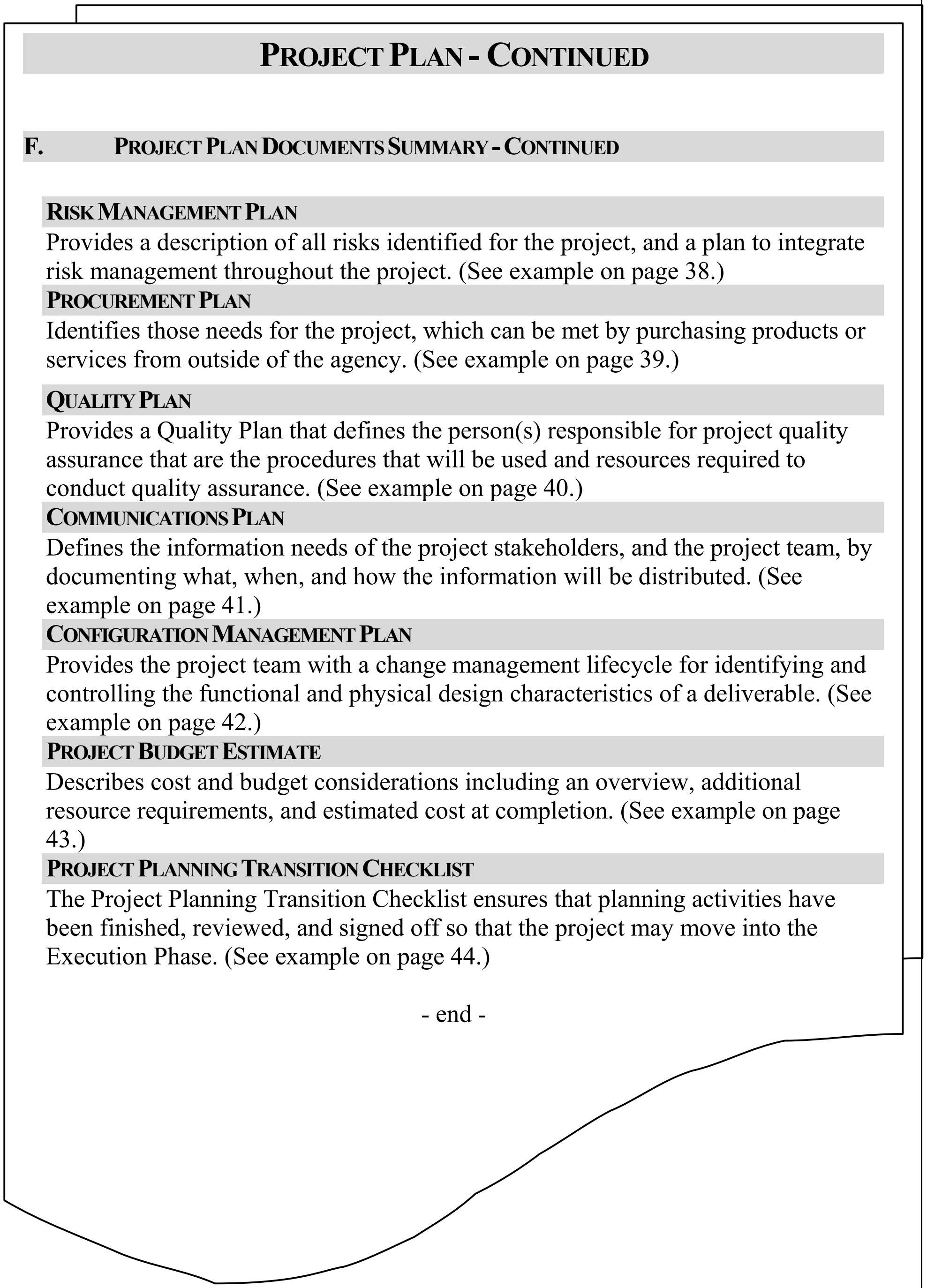
RESOURCE PLAN

Describes the major resources that will be needed to proceed with the execution of the project. (See example on page 36.)

PROJECT SCHEDULE

Provides the project schedule using a Gantt chart. The schedule must include milestones, task dependencies, task duration, work product delivery dates, quality milestones, configuration management milestones, and action items. (See example on page 37.)

- continued -



PROJECT PLAN - CONTINUED

F. PROJECT PLAN DOCUMENTS SUMMARY - CONTINUED

RISK MANAGEMENT PLAN

Provides a description of all risks identified for the project, and a plan to integrate risk management throughout the project. (See example on page 38.)

PROCUREMENT PLAN

Identifies those needs for the project, which can be met by purchasing products or services from outside of the agency. (See example on page 39.)

QUALITY PLAN

Provides a Quality Plan that defines the person(s) responsible for project quality assurance that are the procedures that will be used and resources required to conduct quality assurance. (See example on page 40.)

COMMUNICATIONS PLAN

Defines the information needs of the project stakeholders, and the project team, by documenting what, when, and how the information will be distributed. (See example on page 41.)

CONFIGURATION MANAGEMENT PLAN

Provides the project team with a change management lifecycle for identifying and controlling the functional and physical design characteristics of a deliverable. (See example on page 42.)

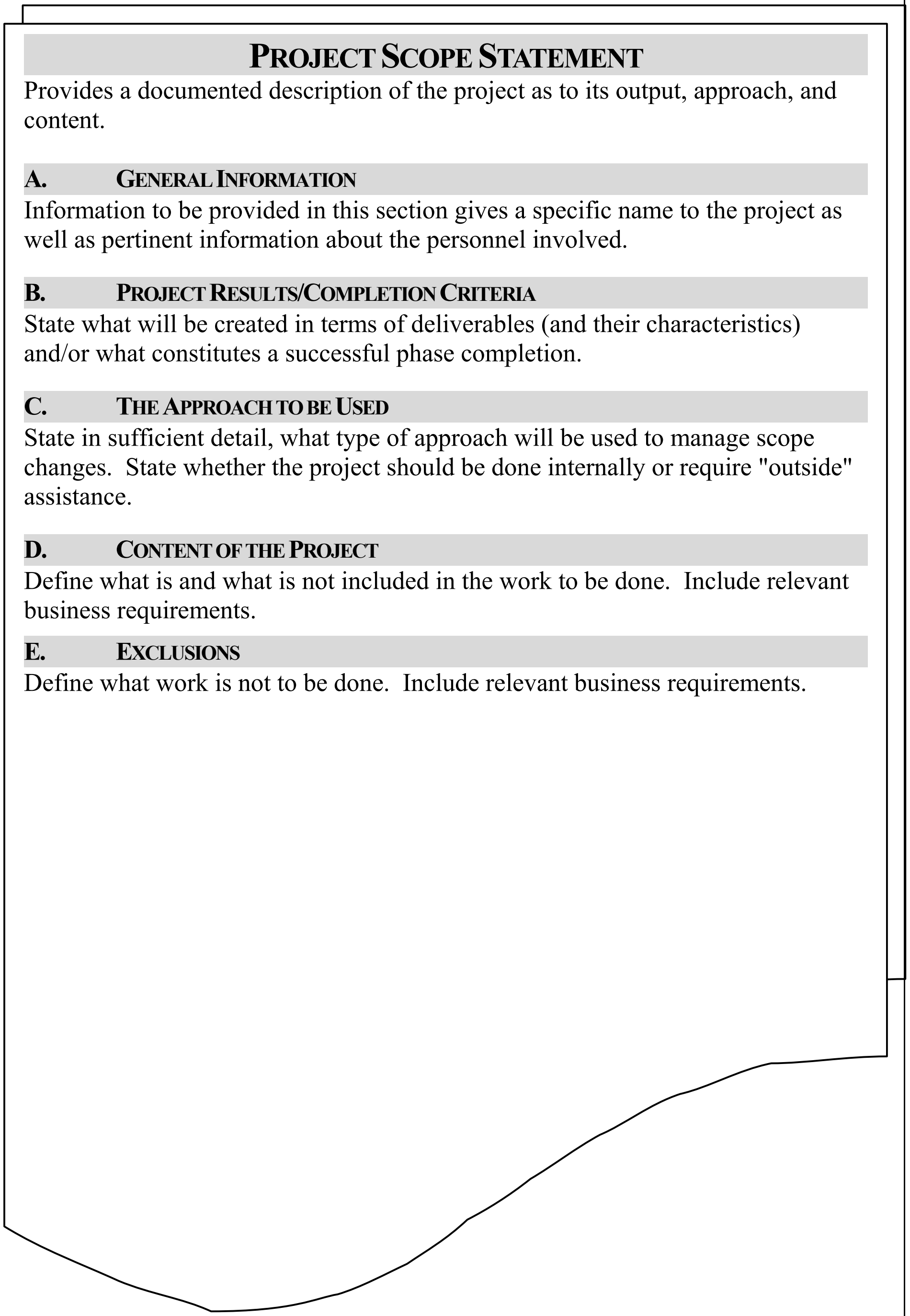
PROJECT BUDGET ESTIMATE

Describes cost and budget considerations including an overview, additional resource requirements, and estimated cost at completion. (See example on page 43.)

PROJECT PLANNING TRANSITION CHECKLIST

The Project Planning Transition Checklist ensures that planning activities have been finished, reviewed, and signed off so that the project may move into the Execution Phase. (See example on page 44.)

- end -



PROJECT SCOPE STATEMENT

Provides a documented description of the project as to its output, approach, and content.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. PROJECT RESULTS/COMPLETION CRITERIA

State what will be created in terms of deliverables (and their characteristics) and/or what constitutes a successful phase completion.

C. THE APPROACH TO BE USED

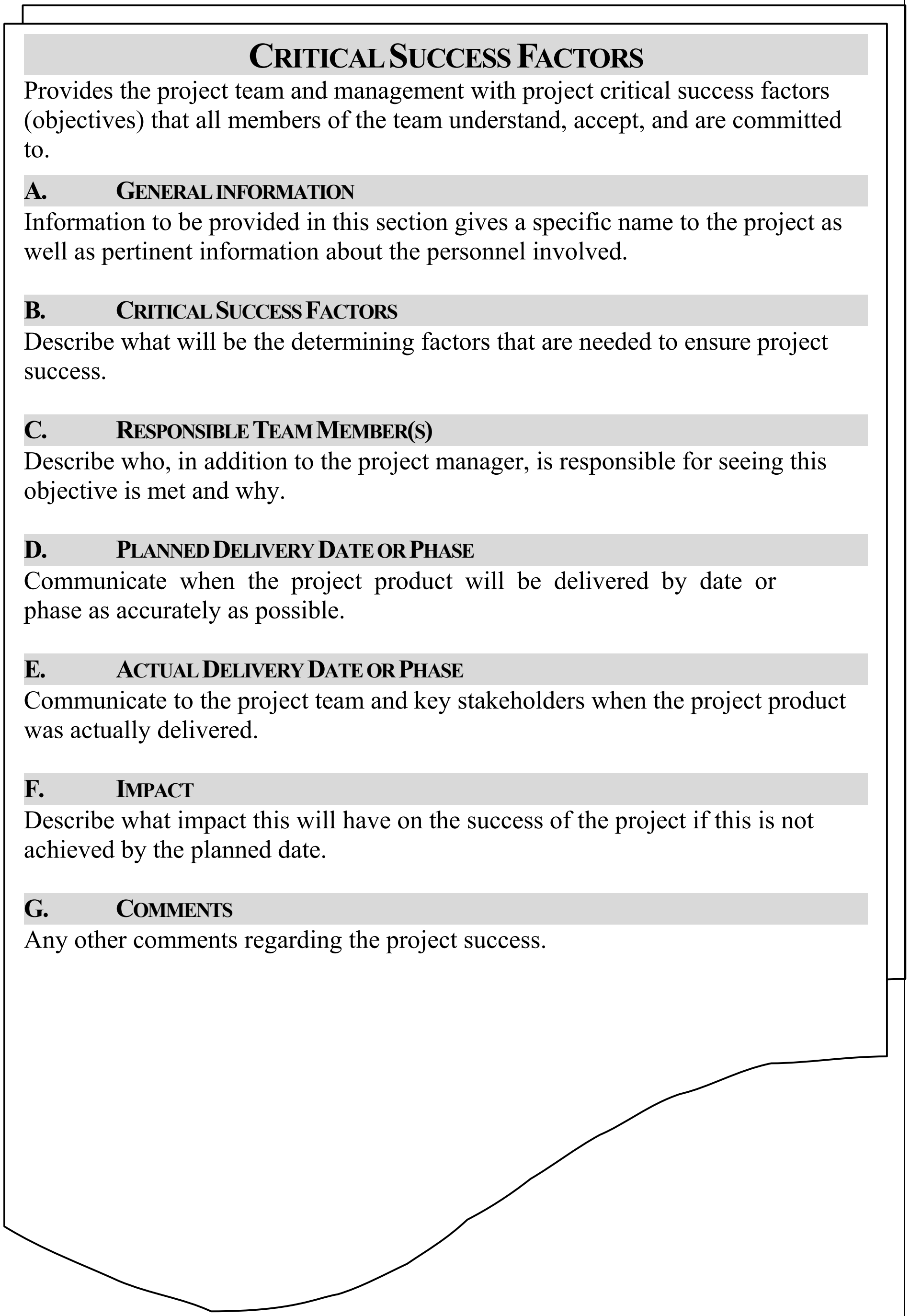
State in sufficient detail, what type of approach will be used to manage scope changes. State whether the project should be done internally or require "outside" assistance.

D. CONTENT OF THE PROJECT

Define what is and what is not included in the work to be done. Include relevant business requirements.

E. EXCLUSIONS

Define what work is not to be done. Include relevant business requirements.



CRITICAL SUCCESS FACTORS

Provides the project team and management with project critical success factors (objectives) that all members of the team understand, accept, and are committed to.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. CRITICAL SUCCESS FACTORS

Describe what will be the determining factors that are needed to ensure project success.

C. RESPONSIBLE TEAM MEMBER(S)

Describe who, in addition to the project manager, is responsible for seeing this objective is met and why.

D. PLANNED DELIVERY DATE OR PHASE

Communicate when the project product will be delivered by date or phase as accurately as possible.

E. ACTUAL DELIVERY DATE OR PHASE

Communicate to the project team and key stakeholders when the project product was actually delivered.

F. IMPACT

Describe what impact this will have on the success of the project if this is not achieved by the planned date.

G. COMMENTS

Any other comments regarding the project success.

WORK BREAKDOWN STRUCTURE

Describes a deliverable-oriented grouping of project elements that organizes and defines the total scope of the project.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. WBS ELEMENT/ACTIVITY NUMBER

Assign a particular number as a part of a sequence related to a certain task or activity.

C. ACTIVITY/TASK NAME

Establish the title of the task or activity.

D. EFFORT/DURATION

Determine the number of effort hours and/or days/weeks it will take to complete the task or activity.

E. RESOURCE NAMES

Assign a responsible team or person for completing the task.

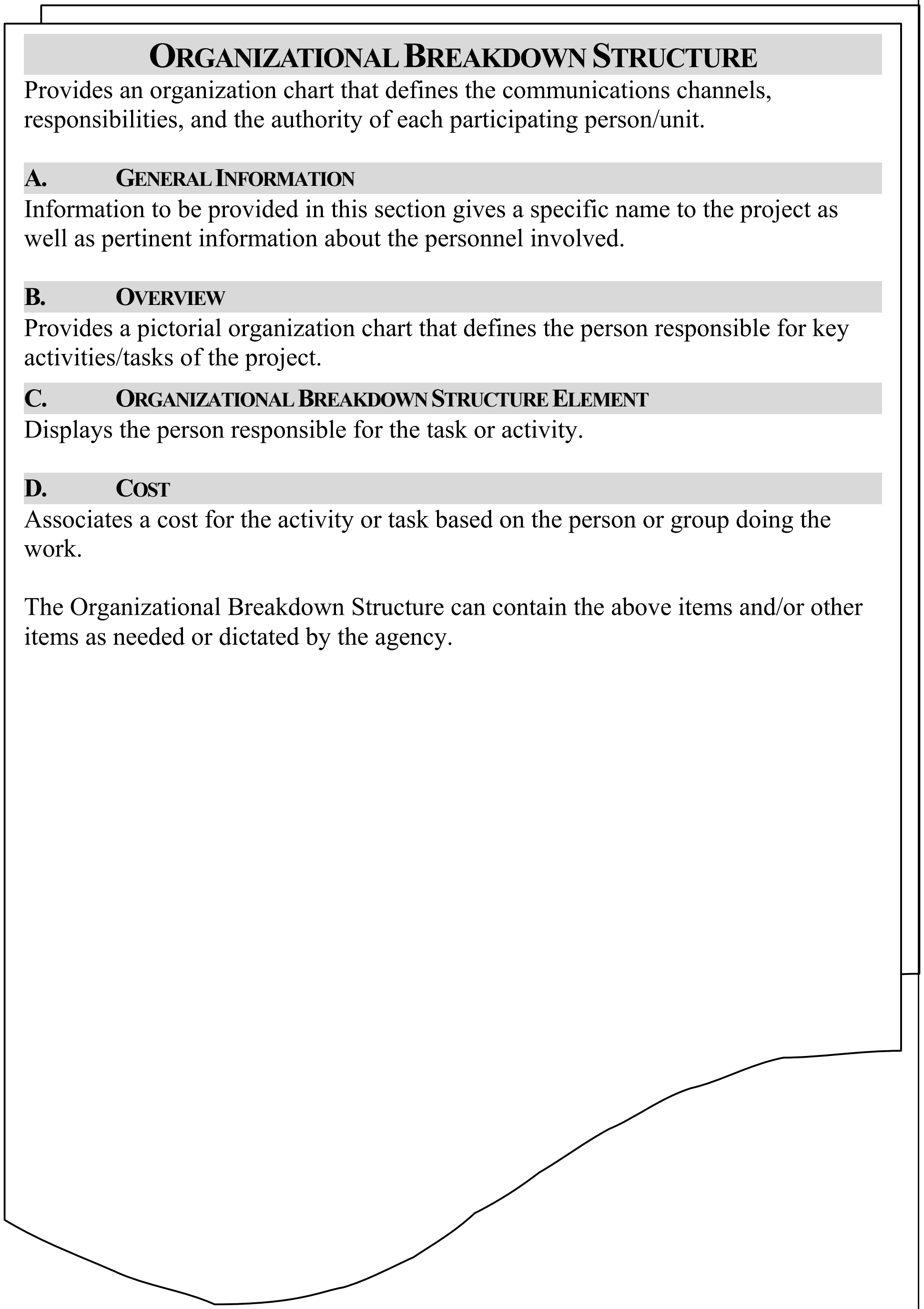
F. DICTIONARY DESCRIPTION

Provide a description of the activity or task.

G. COST

Estimate the cost of performing the activity or task.

The Work Breakdown Structure can contain other elements such as start date, Earned Value Analysis data, dependencies, etc.



ORGANIZATIONAL BREAKDOWN STRUCTURE

Provides an organization chart that defines the communications channels, responsibilities, and the authority of each participating person/unit.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. OVERVIEW

Provides a pictorial organization chart that defines the person responsible for key activities/tasks of the project.

C. ORGANIZATIONAL BREAKDOWN STRUCTURE ELEMENT

Displays the person responsible for the task or activity.

D. COST

Associates a cost for the activity or task based on the person or group doing the work.

The Organizational Breakdown Structure can contain the above items and/or other items as needed or dictated by the agency.

Cost Benefit Analysis

Provides the project team with information to make a balanced decision about the costs and benefits, or value, of various economic choices on proposed project activities/tasks.

A. General information

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. Project Overview and Background

Provide a brief overview, background, and definition of the project

C. Discussion of Alternatives

Discuss the project ground rules and assumptions. These include:

- Status Quo—Current Process (As-Is Model)
- Discussion of Alternative Concepts and Goals
    - Program Concept
    - Functional Concept
    - Technical Concept
- Project Alternatives (To-Be Model)
- Acquisition Strategy
- Discussion of Alternatives
- Schedule

D. Life Cycle Costs and Benefits

Discuss the costs and benefits of the product according to its life cycle. This will include:

- Life Cycle Cost Summary
- Life Cycle Benefit Summary
- Life Cycle Risk Summary
- Sensitivity Analysis
- Life Cycle Cost-Benefit Comparison

RESOURCE PLAN

Describes the major resources that will be needed to proceed with the execution of the project.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. RESOURCE PROFILES

Determine the major resources that will be needed to proceed with the execution of the project. These resources may include the following:

- People, including various skill types
- Money
- Equipment
- Facilities
- Materials and Supplies
- Information Technology

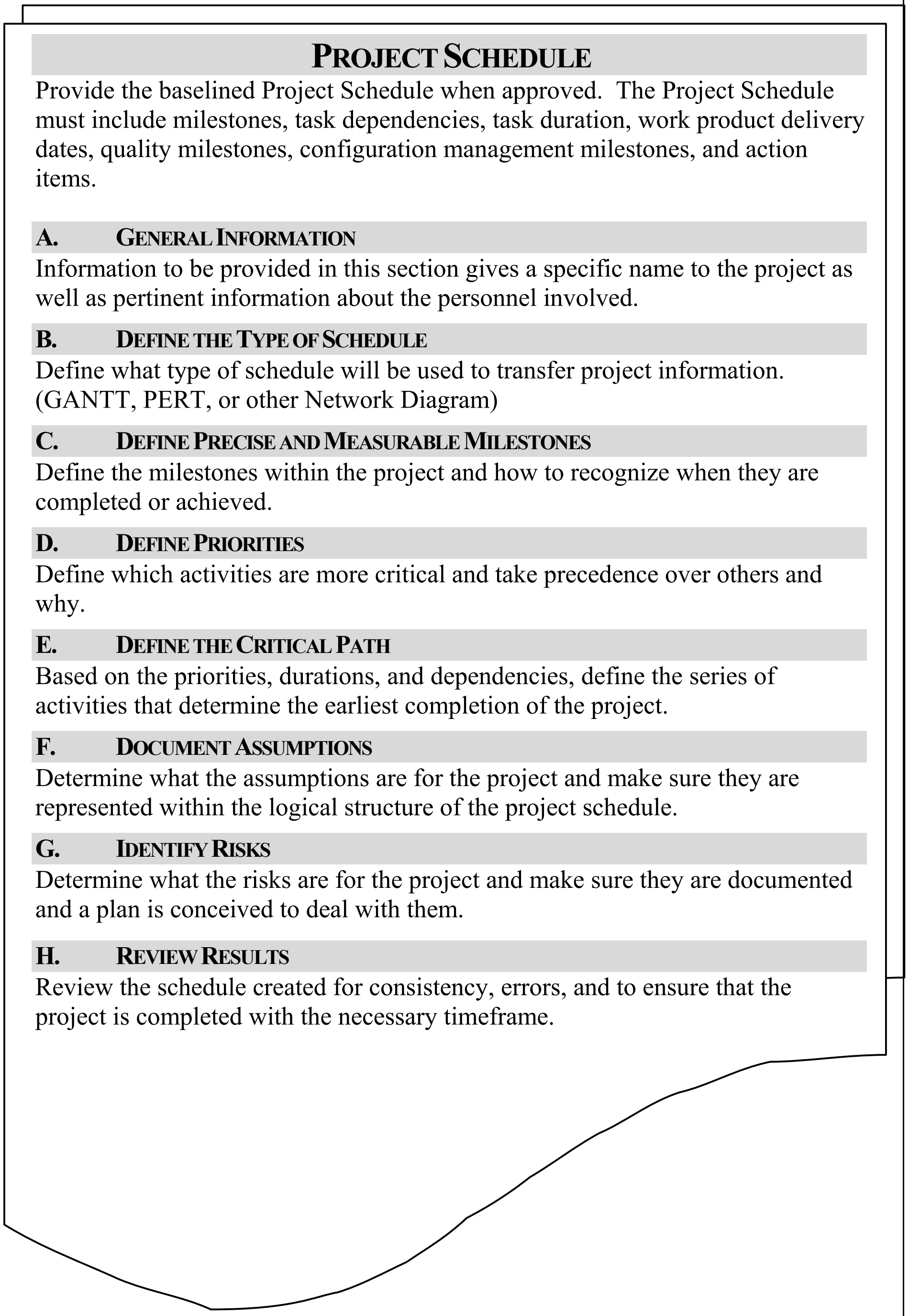
C. PROJECT RESOURCE INFORMATION

For the resources needed on the project, determine the following:

- Cost estimates for each resource.
- The availability of each resource.
- The estimated quality and output of people and equipment resources.

D. RESOURCE STAFFING PLAN

After establishing the human resources required for the project, develop a staffing plan that shows the number of personnel, by type, that will be required on the project on a periodic basis. Include when and how resources – including human, hardware, facilities, etc. – will be phased off the project.



PROJECT SCHEDULE

Provide the baselined Project Schedule when approved. The Project Schedule must include milestones, task dependencies, task duration, work product delivery dates, quality milestones, configuration management milestones, and action items.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. DEFINE THE TYPE OF SCHEDULE

Define what type of schedule will be used to transfer project information. (GANTT, PERT, or other Network Diagram)

C. DEFINE PRECISE AND MEASURABLE MILESTONES

Define the milestones within the project and how to recognize when they are completed or achieved.

D. DEFINE PRIORITIES

Define which activities are more critical and take precedence over others and why.

E. DEFINE THE CRITICAL PATH

Based on the priorities, durations, and dependencies, define the series of activities that determine the earliest completion of the project.

F. DOCUMENT ASSUMPTIONS

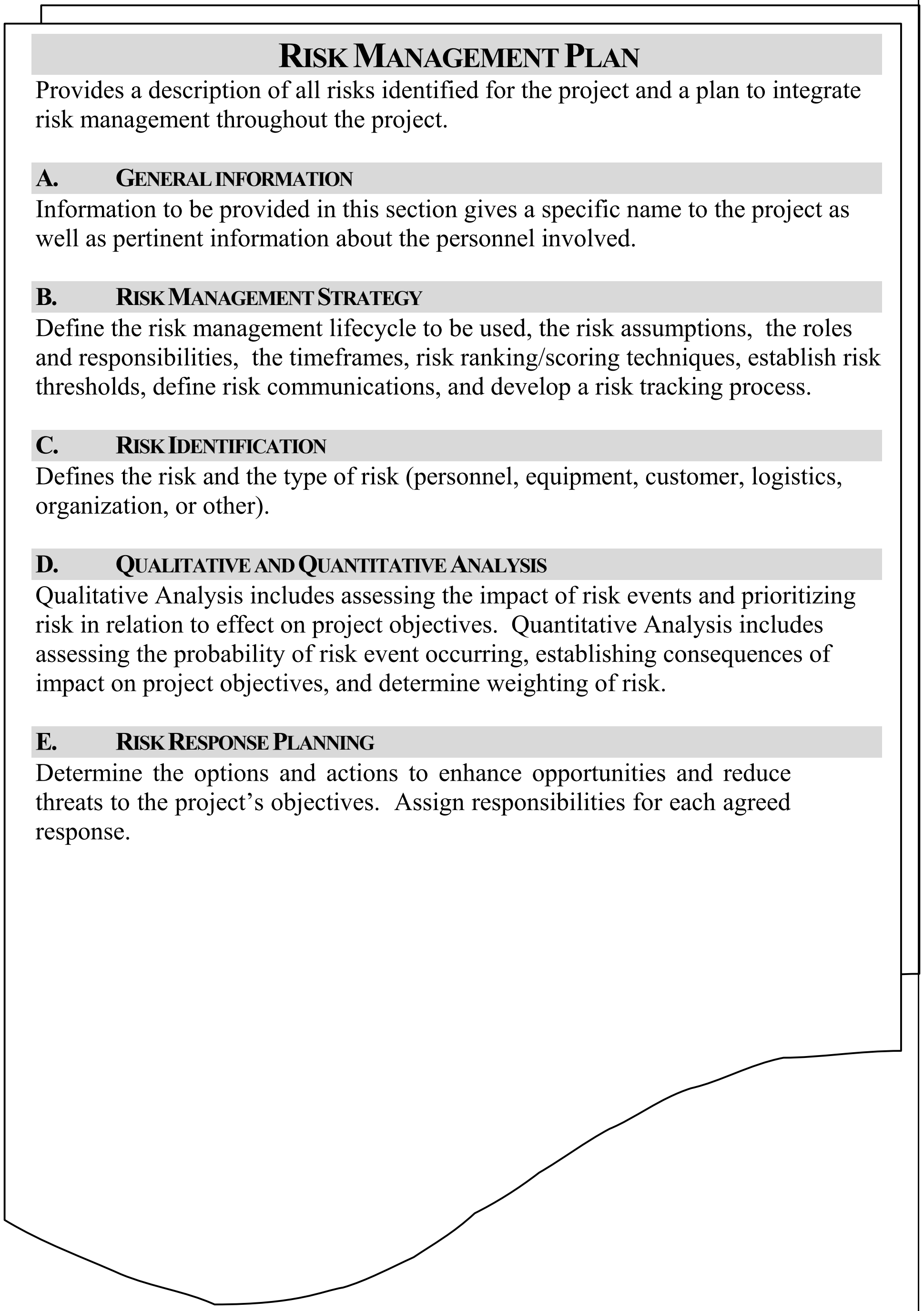
Determine what the assumptions are for the project and make sure they are represented within the logical structure of the project schedule.

G. IDENTIFY RISKS

Determine what the risks are for the project and make sure they are documented and a plan is conceived to deal with them.

H. REVIEW RESULTS

Review the schedule created for consistency, errors, and to ensure that the project is completed with the necessary timeframe.



RISK MANAGEMENT PLAN

Provides a description of all risks identified for the project and a plan to integrate risk management throughout the project.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. RISK MANAGEMENT STRATEGY

Define the risk management lifecycle to be used, the risk assumptions, the roles and responsibilities, the timeframes, risk ranking/scoring techniques, establish risk thresholds, define risk communications, and develop a risk tracking process.

C. RISK IDENTIFICATION

Defines the risk and the type of risk (personnel, equipment, customer, logistics, organization, or other).

D. QUALITATIVE AND QUANTITATIVE ANALYSIS

Qualitative Analysis includes assessing the impact of risk events and prioritizing risk in relation to effect on project objectives. Quantitative Analysis includes assessing the probability of risk event occurring, establishing consequences of impact on project objectives, and determine weighting of risk.

E. RISK RESPONSE PLANNING

Determine the options and actions to enhance opportunities and reduce threats to the project's objectives. Assign responsibilities for each agreed response.

PROCUREMENT PLAN

Identifies those needs for the project that can be met by purchasing products or services from outside of the agency.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. PROCUREMENT DEFINITION

Define in specific terms what items will be procured and under what conditions.

C. CONTRACT RESPONSIBILITY

Define who within the agency will be allowed to enter into contract agreements.

D. DECISION CRITERIA

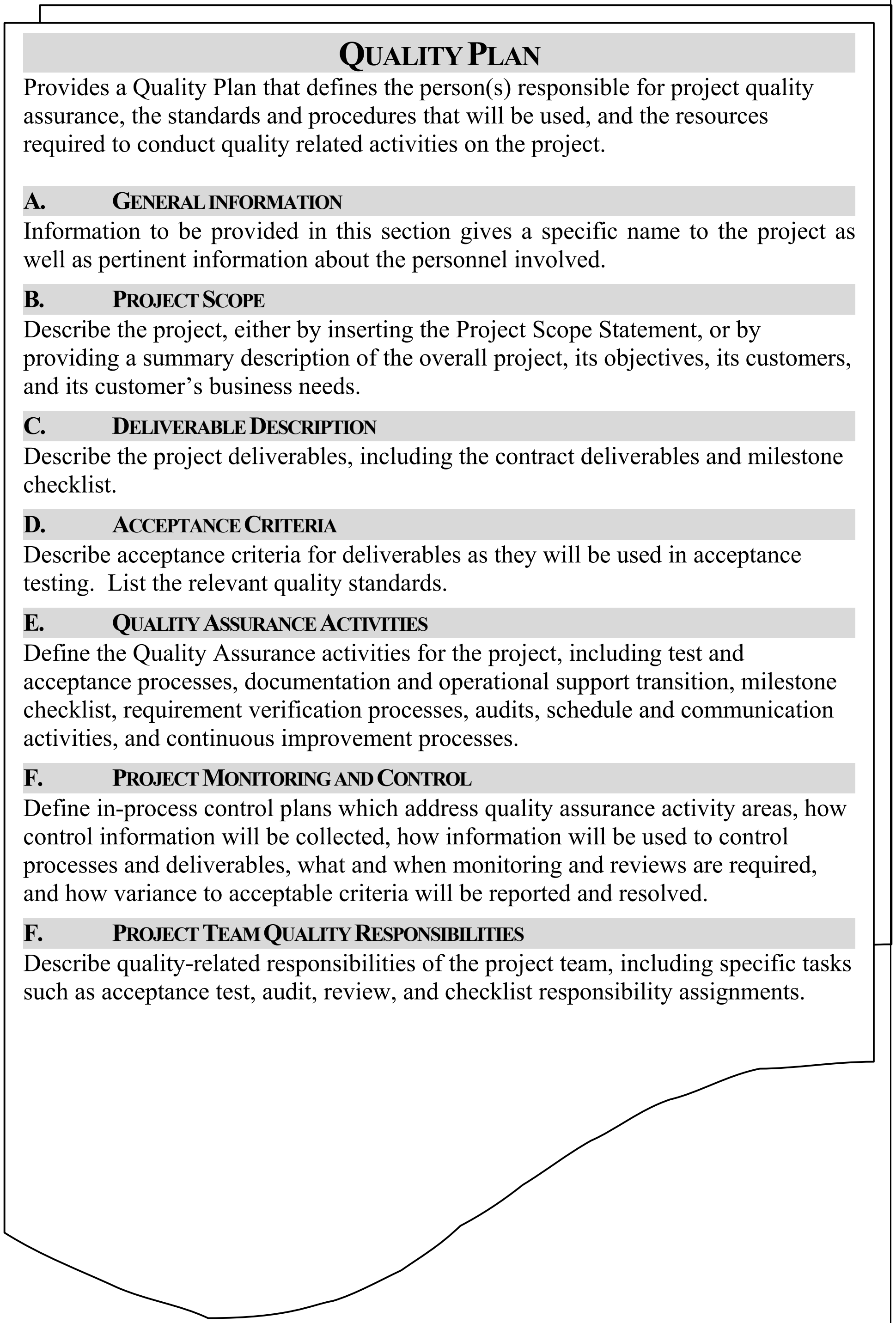
Define what type of analysis will be used to determine make or buy decisions.

E. CONTRACT TYPE

Document what types of contracts will be used and what actions need to be taken to initiate procurement.

F. CONTRACT STANDARDS

Provide the standards for documentation that will need to be initiated and maintained for each contract.



QUALITY PLAN

Provides a Quality Plan that defines the person(s) responsible for project quality assurance, the standards and procedures that will be used, and the resources required to conduct quality related activities on the project.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. PROJECT SCOPE

Describe the project, either by inserting the Project Scope Statement, or by providing a summary description of the overall project, its objectives, its customers, and its customer's business needs.

C. DELIVERABLE DESCRIPTION

Describe the project deliverables, including the contract deliverables and milestone checklist.

D. ACCEPTANCE CRITERIA

Describe acceptance criteria for deliverables as they will be used in acceptance testing. List the relevant quality standards.

E. QUALITY ASSURANCE ACTIVITIES

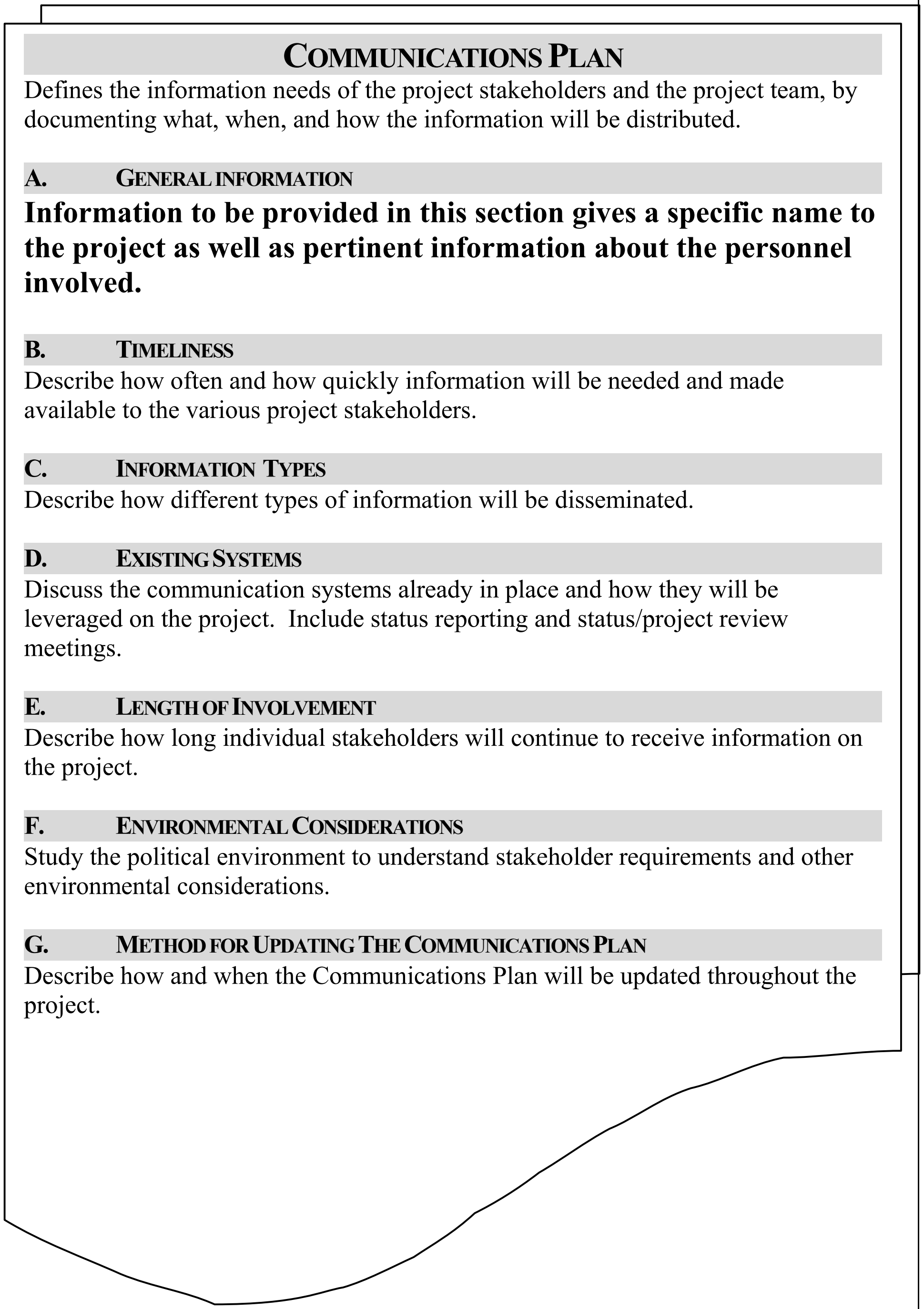
Define the Quality Assurance activities for the project, including test and acceptance processes, documentation and operational support transition, milestone checklist, requirement verification processes, audits, schedule and communication activities, and continuous improvement processes.

F. PROJECT MONITORING AND CONTROL

Define in-process control plans which address quality assurance activity areas, how control information will be collected, how information will be used to control processes and deliverables, what and when monitoring and reviews are required, and how variance to acceptable criteria will be reported and resolved.

F. PROJECT TEAM QUALITY RESPONSIBILITIES

Describe quality-related responsibilities of the project team, including specific tasks such as acceptance test, audit, review, and checklist responsibility assignments.



COMMUNICATIONS PLAN

Defines the information needs of the project stakeholders and the project team, by documenting what, when, and how the information will be distributed.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. TIMELINESS

Describe how often and how quickly information will be needed and made available to the various project stakeholders.

C. INFORMATION TYPES

Describe how different types of information will be disseminated.

D. EXISTING SYSTEMS

Discuss the communication systems already in place and how they will be leveraged on the project. Include status reporting and status/project review meetings.

E. LENGTH OF INVOLVEMENT

Describe how long individual stakeholders will continue to receive information on the project.

F. ENVIRONMENTAL CONSIDERATIONS

Study the political environment to understand stakeholder requirements and other environmental considerations.

G. METHOD FOR UPDATING THE COMMUNICATIONS PLAN

Describe how and when the Communications Plan will be updated throughout the project.

Configuration Management Plan

Provides the project team with a change management lifecycle for identifying and controlling the functional and physical design characteristics of a deliverable.

A. General information

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. Configuration Management Functional Area and Resources

Describe the Configuration Management organization structure, personnel skill level and qualifications, facilities needed, and equipment and tools used.

C. Standards, Procedures, Policies, and Guidelines

Display the diagram of information flow and the parameters for any automated tool sets.

D. Configuration Identification

Describe the method for defining each control item, the method for configuration control, and the list of control items.

E. Identification Methods

Describe the naming and marking of documents, components, revisions, releases, etc.

F. Submission and Retrieval of Control Items

Describe the process for submission and retrieval of controlled items within the project.

G. Version Control

Describe the preparation of documentation versions and the release approval procedure.

H. Storage Handling and Delivery of Project Media

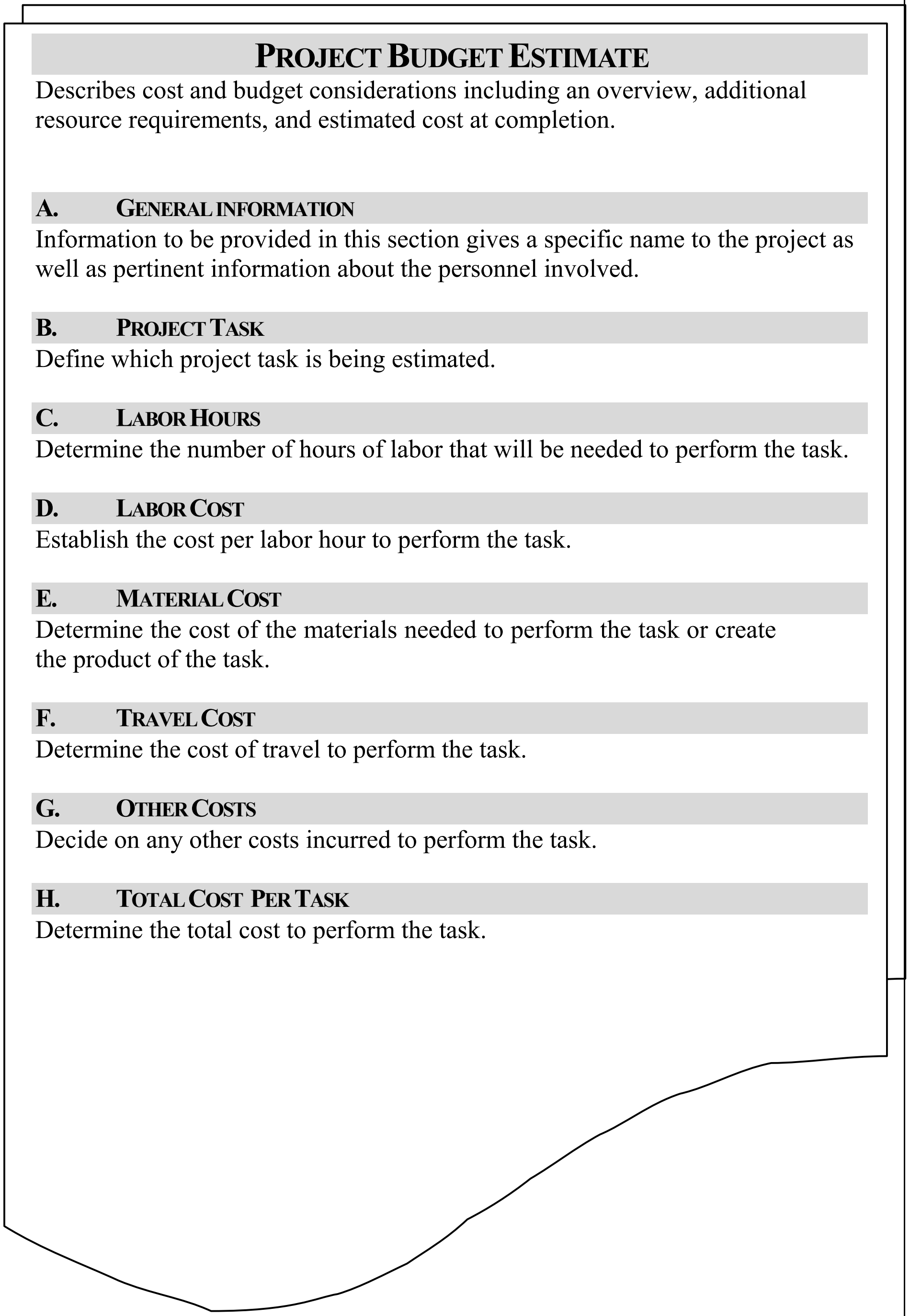
Describe storage requirements (both automated and paper).

I. Relationship to Contractor Configuration Management

Describe the relationship of the Configuration Management team to other Configuration Management teams related to the project.

J. Other information

Relay any other pertinent information about Configuration Management.



PROJECT BUDGET ESTIMATE

Describes cost and budget considerations including an overview, additional resource requirements, and estimated cost at completion.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. PROJECT TASK

Define which project task is being estimated.

C. LABOR HOURS

Determine the number of hours of labor that will be needed to perform the task.

D. LABOR COST

Establish the cost per labor hour to perform the task.

E. MATERIAL COST

Determine the cost of the materials needed to perform the task or create the product of the task.

F. TRAVEL COST

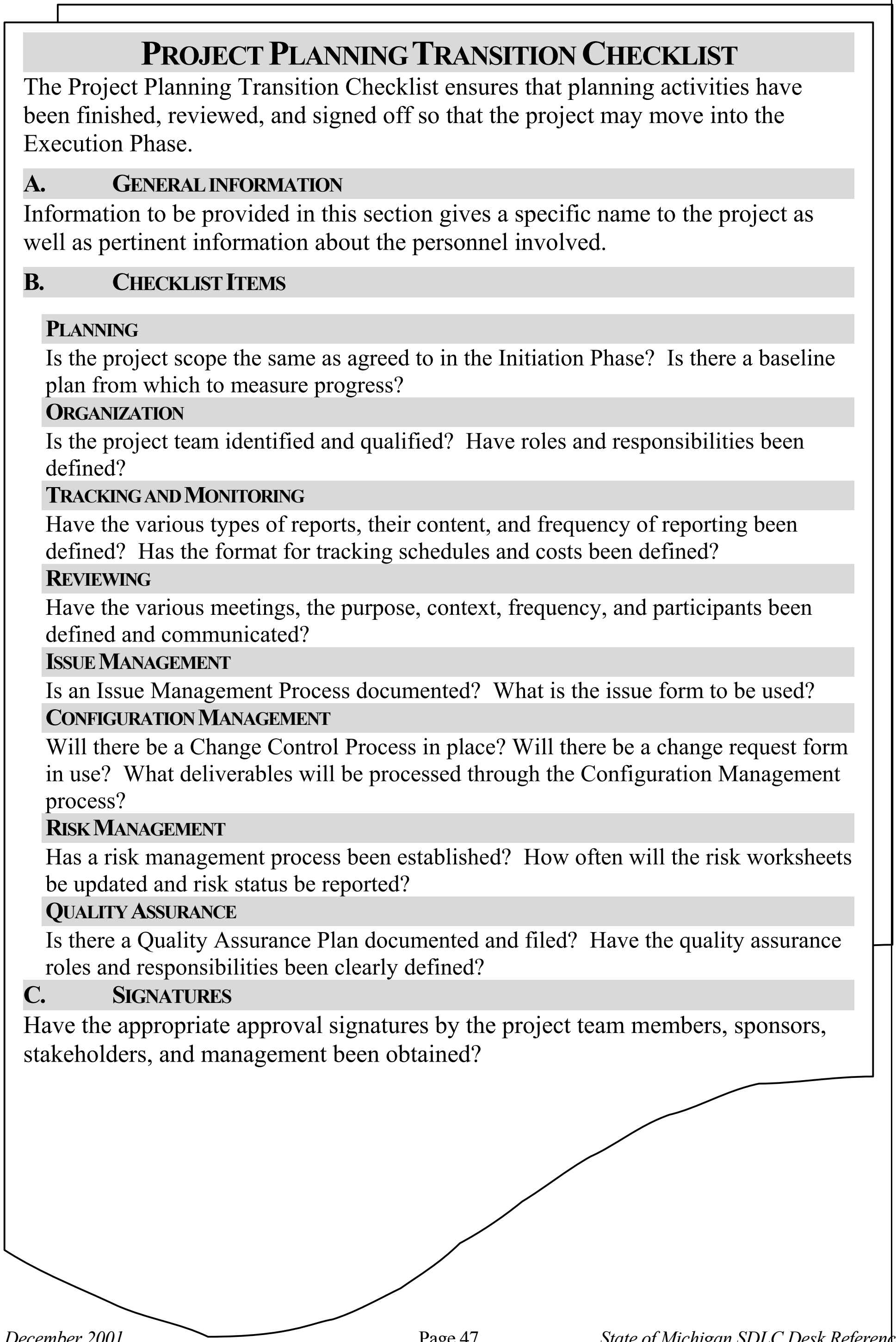
Determine the cost of travel to perform the task.

G. OTHER COSTS

Decide on any other costs incurred to perform the task.

H. TOTAL COST PER TASK

Determine the total cost to perform the task.



PROJECT PLANNING TRANSITION CHECKLIST

The Project Planning Transition Checklist ensures that planning activities have been finished, reviewed, and signed off so that the project may move into the Execution Phase.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. CHECKLIST ITEMS

PLANNING

Is the project scope the same as agreed to in the Initiation Phase? Is there a baseline plan from which to measure progress?

ORGANIZATION

Is the project team identified and qualified? Have roles and responsibilities been defined?

TRACKING AND MONITORING

Have the various types of reports, their content, and frequency of reporting been defined? Has the format for tracking schedules and costs been defined?

REVIEWING

Have the various meetings, the purpose, context, frequency, and participants been defined and communicated?

ISSUE MANAGEMENT

Is an Issue Management Process documented? What is the issue form to be used?

CONFIGURATION MANAGEMENT

Will there be a Change Control Process in place? Will there be a change request form in use? What deliverables will be processed through the Configuration Management process?

RISK MANAGEMENT

Has a risk management process been established? How often will the risk worksheets be updated and risk status be reported?

QUALITY ASSURANCE

Is there a Quality Assurance Plan documented and filed? Have the quality assurance roles and responsibilities been clearly defined?

C. SIGNATURES

Have the appropriate approval signatures by the project team members, sponsors, stakeholders, and management been obtained?

Notes

Requirements Definition Phase

Requirements Definition Phase

Inputs:

- □ Project Notebook
- □ Description of customer
- □ Statement of project scope and
- □ Statement of high-level project
- □ Functional area contact list and project
- □ Summary of platform
- □ Statement of project
- □ Cost Benefit
- □ Feasibility Study
- □ Project Plan
- □ Software Quality Assurance
- □ Software Configuration Management
- □ Risk Plan
- □ Communications

Outputs:

- □ Description of analysis
- □ Records of all project
- □ Customer-oriented requirements *(optional)*
- □ Continuity of Operations
- □ Data Dictionary
- □ Requirements Traceability
- □ Software Requirements
- □ Project Test Plan
- □ Acceptance Test Plan *(draft)*
- □ Design
- □ Project Plan *(revised)*

Key Activities:

Requirements
Select Requirements Analysis
Define Project
Compile and Document Project
Establish Functional
Develop Project Test
Develop Acceptance Test
Select Design
Revise Project Plan
Conduct Structured
Conduct In-Phase
Conduct Requirements Definition Phase

Methods and Tools

Structured

Peer Reviews

In-Phase

Quality

Phase Exit

Forms Utilized

Requirements Traceability
Software Requirements
Software Change Control Request

Software Change Control
Continuity Operations
Project Test Plans, Acceptance Test

The primary goal of the requirement definition phase is to develop a basis of mutual understanding between the system owner/customers and the project team about the requirements for the project. The result of this understanding is an approved Software Requirements Specification that becomes the initial baseline for software product design and a reference for determining whether the completed software product performs as the system owner requested and expected.

This phase involves analysis of the system owner/customers' business processes and needs, translation of those processes and needs into formal requirements, and planning the testing activities to validate the performance of the software product.

PROJECT MANAGER ROLES AND RESPONSIBILITIES

1) Develop a basis of mutual understanding of project requirements;
2) Obtain an approved software requirements specification;
3) Analyze customers business processes and needs;
4) Develop formal requirements document;
5) Plan test activities

ACCEPTANCE TEST PLAN

The Acceptance Test Plan is used to describe the purpose of the test plan and the organization of the document.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. PROJECT /SYSTEM INFORMATION

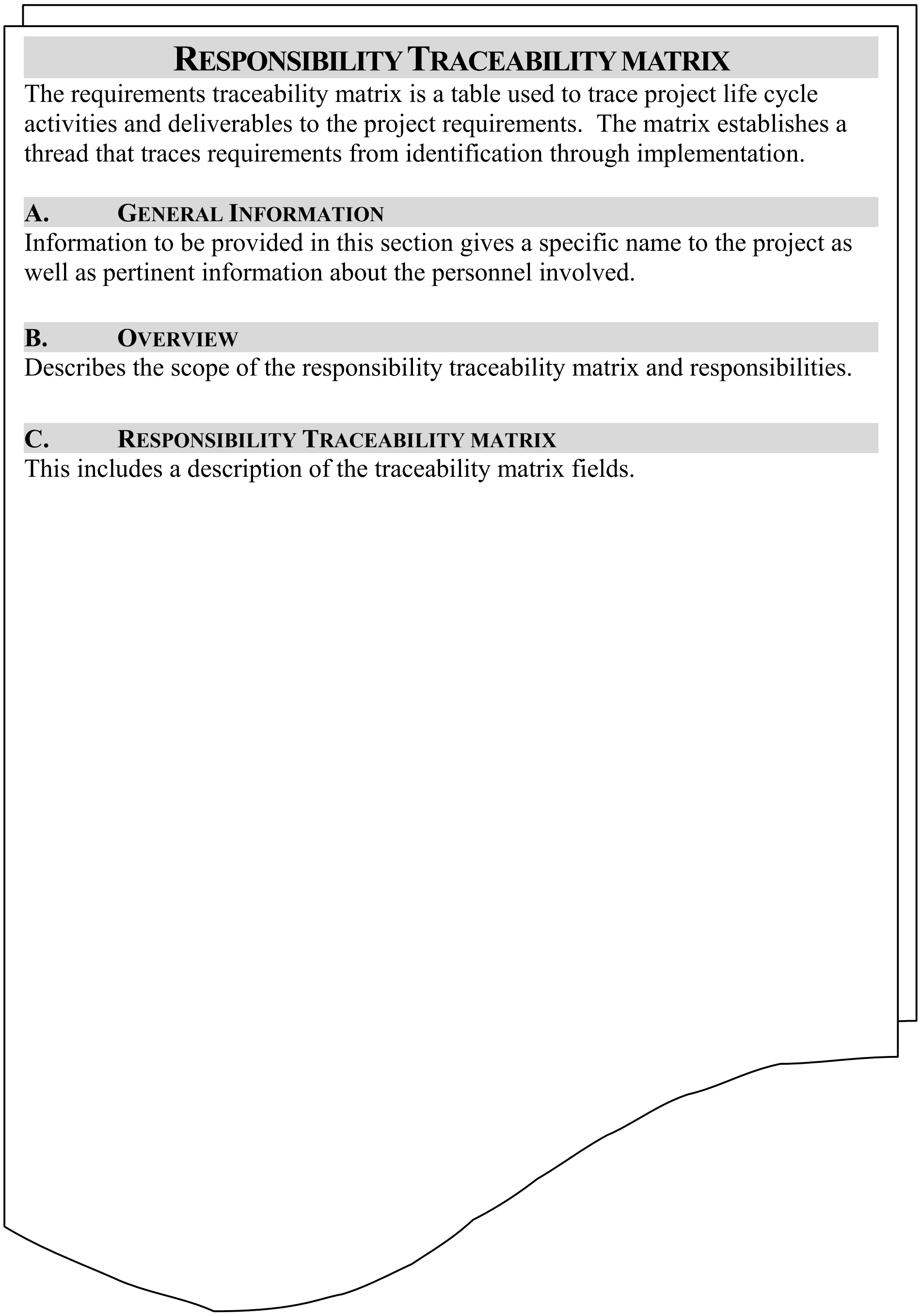
Describes the purpose of the acceptance test plan and the organization of the document. This includes the project objectives, system description, references, outstanding issues and roles and responsibilities.

C. TEST PLAN

This includes the scope of the acceptance test, testing approach, test schedule, problem reporting and data recording, resource requirements, test environment, identification of tests, acceptance test report, and corrective action.

D. TEST CASES

Sample test cases and scenarios are provided.



RESPONSIBILITY TRACEABILITY MATRIX

The requirements traceability matrix is a table used to trace project life cycle activities and deliverables to the project requirements. The matrix establishes a thread that traces requirements from identification through implementation.

A. GENERAL INFORMATION

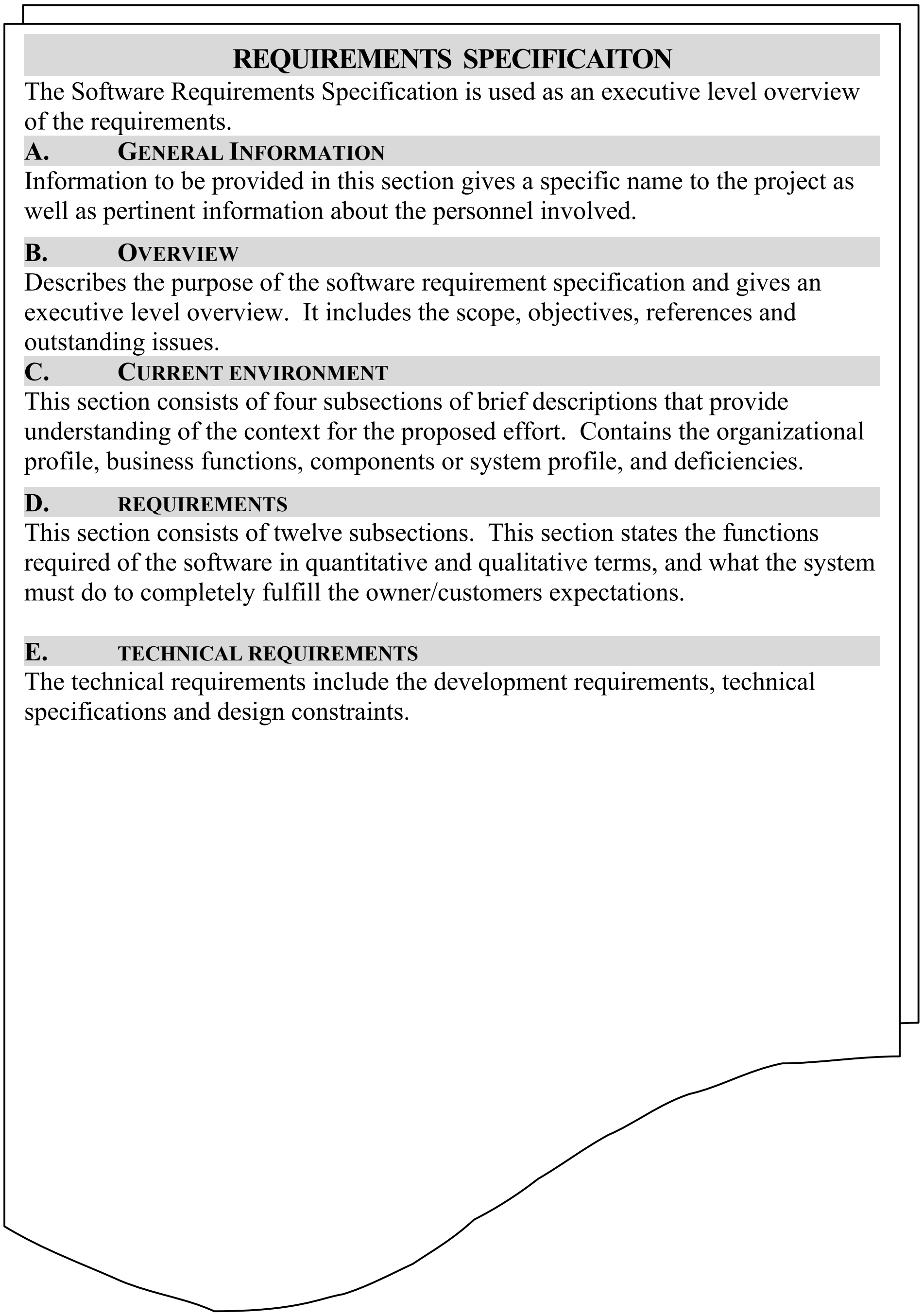
Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. OVERVIEW

Describes the scope of the responsibility traceability matrix and responsibilities.

C. RESPONSIBILITY TRACEABILITY MATRIX

This includes a description of the traceability matrix fields.



REQUIREMENTS SPECIFICAITON

The Software Requirements Specification is used as an executive level overview of the requirements.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. OVERVIEW

Describes the purpose of the software requirement specification and gives an executive level overview. It includes the scope, objectives, references and outstanding issues.

C. CURRENT ENVIRONMENT

This section consists of four subsections of brief descriptions that provide understanding of the context for the proposed effort. Contains the organizational profile, business functions, components or system profile, and deficiencies.

D. REQUIREMENTS

This section consists of twelve subsections. This section states the functions required of the software in quantitative and qualitative terms, and what the system must do to completely fulfill the owner/customers expectations.

E. TECHNICAL REQUIREMENTS

The technical requirements include the development requirements, technical specifications and design constraints.

SOFTWARE DEVELOPMENT RISK ASSESSMENT

The purpose of the Software Development Risk Assessment is to provide project managers with a tool for identifying and planning for potential project risks. It is process-based and supports the framework established by the SOM Systems Development Lifecycle. It will be used within the stage exit process as an additional tool to ensure that the project manager has identified and is managing known risk factors. Additional detailed information describes the various risk factors and how to score them.

.

Notes

Functional Design Phase

Functional Design Phase

Inputs:

- Project Notebook
- Software Configuration Management Plan *(draft)*
- Continuity of Operations Statement/Plan
- Data Dictionary
- Requirements Traceability Matrix
- Software Requirements Specification
- Project Test Plan
- Acceptance Test Plan *(revised)*
- Design methodology
- Project Plan *(revised)*
- Software Quality Assurance Plan

Outputs:

- Design records
- Logical model
- Data Dictionary
- Requirements Traceability Matrix *(expanded)*
- Functional Design Document
- Minutes from Functional Design Review
- Hardware and software procurement records

❑ Project Plan *(revised)*

Key Activities:

Determine Software Structure
Design Content of System Inputs and Outputs
Design Customer Interface
Design System Interfaces
Design System Security Controls
Build Logical Model
Build Data Model
Develop Functional Design
Initiate Procurement of Hardware and Software
Revise Project Plan
Conduct Structured Walkthrough(s)
Conduct In-Phase Assessment
Conduct Functional Design Phase Exit

Methods and Tools

Define requirements for:

❑ Data flowing in and out of system - hardcopy, and softcopy
❑ Processes required for the processing of data - manual and automated
❑ Storage required - structure of data, relationships between data and size of files

Choose solution considering:

❑ Technological feasibility
❑ Cost / Benefit analysis
❑ Operational feasibility

Forms Utilized

Functional Design Document

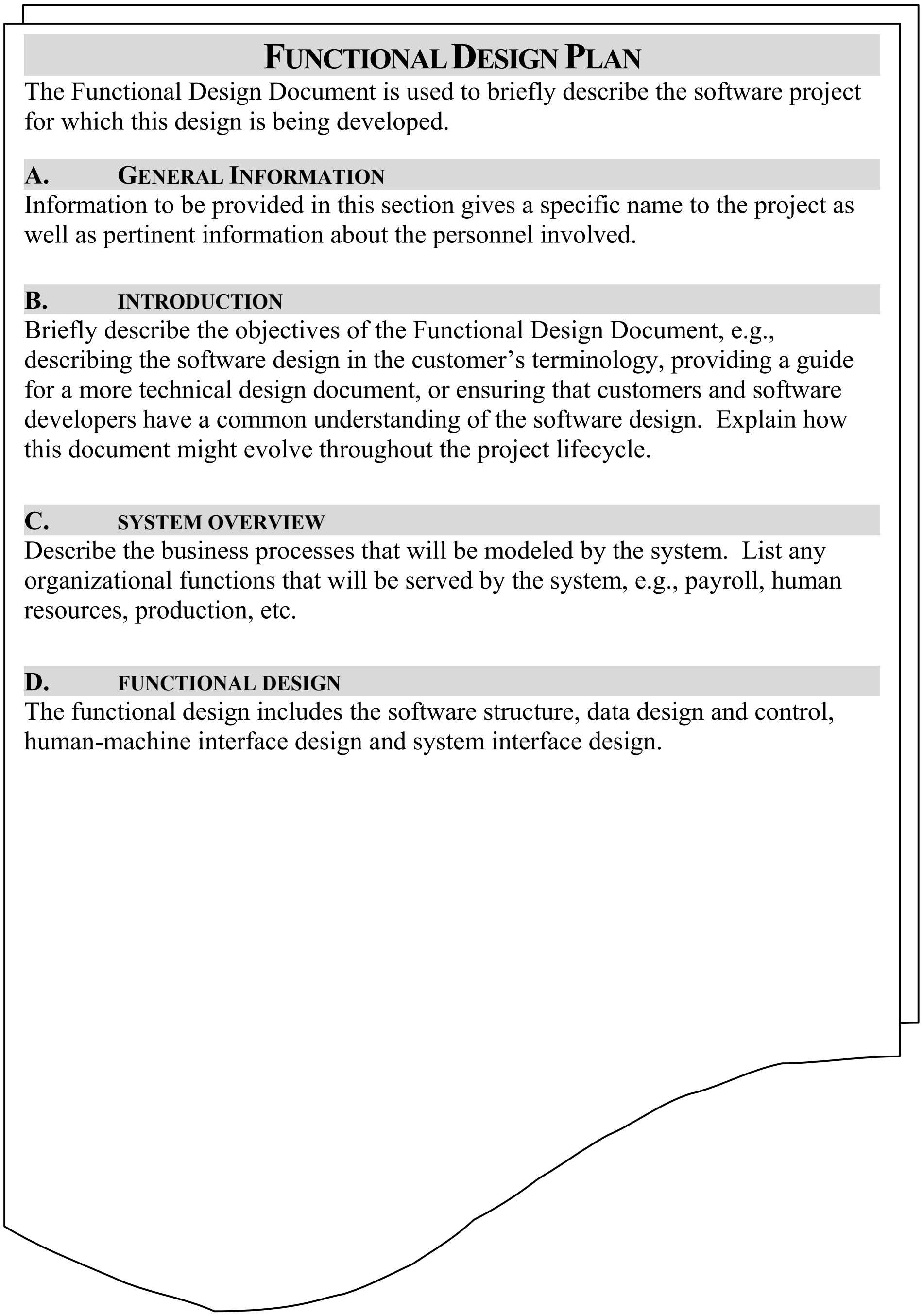
The functional design process maps the "what to do" of the Software Requirements Specification into the "how to do it" of the design specifications. During this phase, the overall structure of the software product is defined from a functional viewpoint. The functional design describes the logical system flow, data organization, system inputs and outputs, processing rules, and operational characteristics of the software product from the customer's point of view. The functional design is not concerned with the software or hardware that will support the operation of the software product, or the physical organization of the data or the programs that will accept the input data, execute the processing rules, and produce the required output.

The focus is on the functions and structure of the components that comprise the software product. The goal of this phase is to define and document the functions of the software product to the extent necessary to obtain the system owner and customers understanding and approval and to the level of detail necessary to build the system design.

Prototyping of system functions can be helpful in communicating the design specifications to the system owner and customers. Prototypes can be used to simulate one function, a module, or the entire software product. Prototyping is also useful in the transition from the functional design to the system design.

PROJECT MANAGER ROLES AND RESPONSIBILITIES

1) *Determine what to do;*
2) *Determine how to do it;*
3) *Define structure of software product;*
4) *Develop logical system flow;*
5) *Data organization;*
6) *Systems inputs;*
7) *System outputs;*
8) *Processing rules;*
9) *Operational characteristics*



FUNCTIONAL DESIGN PLAN

The Functional Design Document is used to briefly describe the software project for which this design is being developed.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. INTRODUCTION

Briefly describe the objectives of the Functional Design Document, e.g., describing the software design in the customer’s terminology, providing a guide for a more technical design document, or ensuring that customers and software developers have a common understanding of the software design. Explain how this document might evolve throughout the project lifecycle.

C. SYSTEM OVERVIEW

Describe the business processes that will be modeled by the system. List any organizational functions that will be served by the system, e.g., payroll, human resources, production, etc.

D. FUNCTIONAL DESIGN

The functional design includes the software structure, data design and control, human-machine interface design and system interface design.

Notes

Systems Design Phase

Systems Design Phase Highlights

Inputs:

- Project Notebook
- Design records
- Logical model
- Data dictionary *(expanded)*
- Requirements Traceability Matrix *(expanded)*
- Functional Design Document
- Hardware and software procurement records
- Project Plan *(revised)*
- Software Quality Assurance Plan

Outputs:

- Design specifications
- Physical Model
- Data Dictionary *(expanded)*
- Integration Test Plan *(draft)*
- System Test Plan *(draft)*
- Conversion Plan
- Requirements Traceability Matrix *(expanded)*
- System Design Document
- Program Specifications
- Programming Standards
- Design Plan *(revised)*

Key Activities:

Select System Architecture
Design Specifications for Software Modules
Design Physical Model and Data Base Structure
Develop Integration Test Plan
Develop System Test Plan
Develop Conversion Plan
Develop System Design
Develop Program Specifications
Define Programming Standards
Revise Project Plan
Conduct Structured Walkthrough(s)
Conduct In-Phase Assessment
Conduct System Design Phase Exit

Forms Utilized

Conversion Plan
System Design Document

Methods and Tools

Define procedures and man-machine boundaries:

❑ Manual and automated processing
❑ Security and quality issues
❑ Input and output functions

Tools:

❑ project management tools
❑ programming code and hardware
❑ prototyping tools
❑ CASE (computer aided software engineering)

Evaluate design

❑ *Walkthrough* procedures being developed
❑ Use feedback from users where possible

Present design documentation

❑ Executive summary of system
❑ Description and explanation of design decisions
❑ Detailed documentation required for programmers, storage requirements etc
❑ Control descriptions

The goal of this phase is to translate the customer-oriented functional design specifications into a set of technical, computer-oriented system design specifications; and to design the data structure and processes to the level of detail necessary to plan and execute the Programming and Installation Phases. General module specifications should be developed to define what each module is to do, but not how the module is to be coded. Effort focuses on specifying individual routines and data structures while holding constant the software structure and interfaces developed in the previous phase. Each module and data structure is considered individually during detailed design with emphasis placed on the description of internal and procedural details. The primary deliverable of this phase is a software system design that provides a blueprint for the coding of individual modules and programs.

PROJECT MANAGER ROLES AND RESPONSIBILITIES

1) *Translate customer-oriented functional design specifications;*
2) *Design the data structure and processes;*
3) *Develop general module specifications;*
4) *Develop system design*

SYSTEMS DESIGN PLAN

The Systems Design Plan is used to describe the system, system owner, and functions of the system.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. INTRODUCTION

Describes the systems objectives, plan objectives, and references.

C. SYSTEM ARCHITECTURE

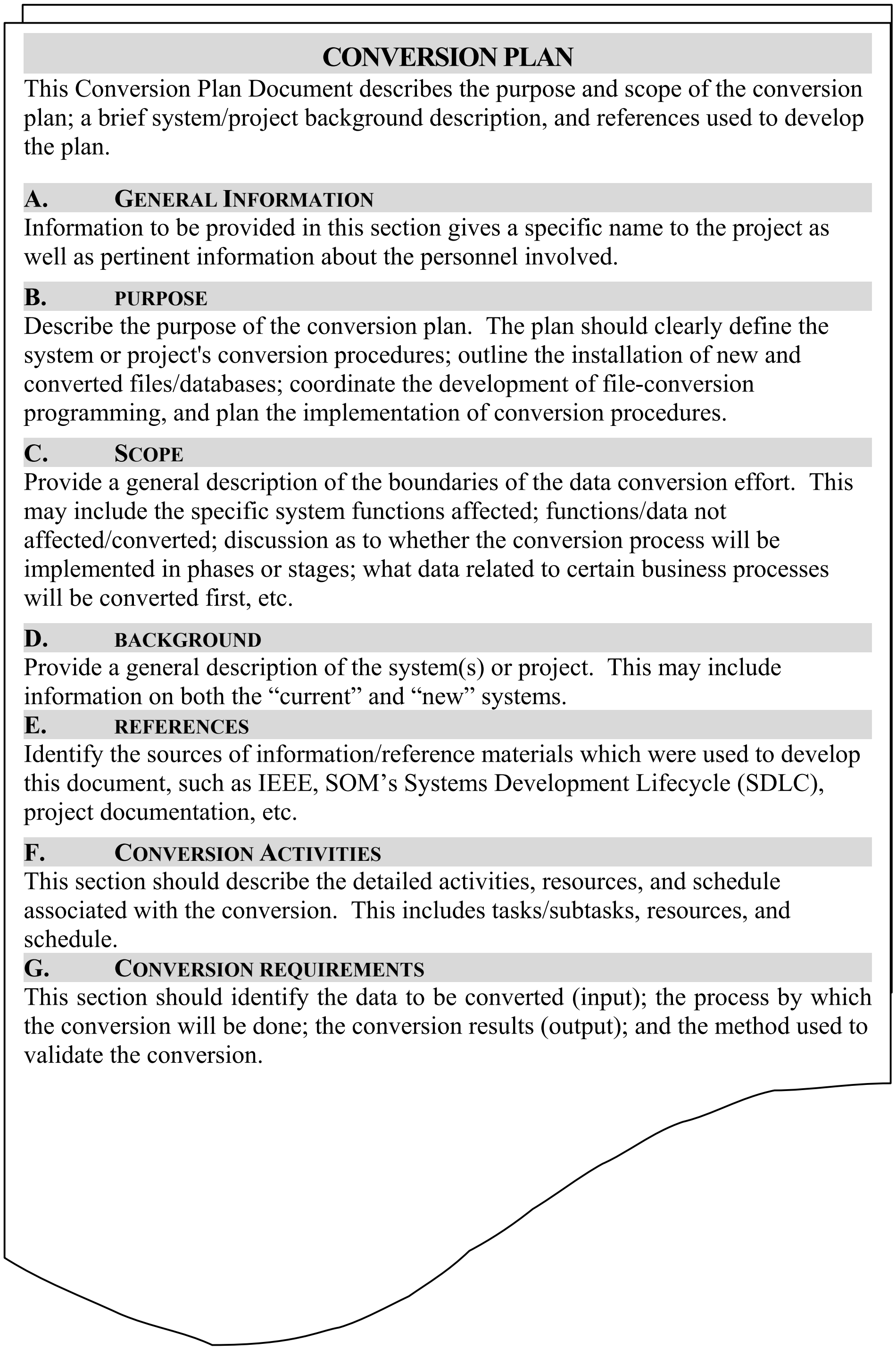
This includes the hardware, data communications, software and architecture diagram.

D. DATA DESIGN

This includes the data objects and resultant data structures and the file and data structures.

E. MODULAR DESIGN

This includes the modules, processing narrative, internal data structures and design language.



CONVERSION PLAN

This Conversion Plan Document describes the purpose and scope of the conversion plan; a brief system/project background description, and references used to develop the plan.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. PURPOSE

Describe the purpose of the conversion plan. The plan should clearly define the system or project's conversion procedures; outline the installation of new and converted files/databases; coordinate the development of file-conversion programming, and plan the implementation of conversion procedures.

C. SCOPE

Provide a general description of the boundaries of the data conversion effort. This may include the specific system functions affected; functions/data not affected/converted; discussion as to whether the conversion process will be implemented in phases or stages; what data related to certain business processes will be converted first, etc.

D. BACKGROUND

Provide a general description of the system(s) or project. This may include information on both the "current" and "new" systems.

E. REFERENCES

Identify the sources of information/reference materials which were used to develop this document, such as IEEE, SOM's Systems Development Lifecycle (SDLC), project documentation, etc.

F. CONVERSION ACTIVITIES

This section should describe the detailed activities, resources, and schedule associated with the conversion. This includes tasks/subtasks, resources, and schedule.

G. CONVERSION REQUIREMENTS

This section should identify the data to be converted (input); the process by which the conversion will be done; the conversion results (output); and the method used to validate the conversion.

Notes

Programming Phase

Programming Phase

Inputs:

- Project Notebook
- Design specifications
- Physical model
- Data Dictionary
- Integration Test Plan *(draft)*
- System Test Plan *(draft)*
- Conversion Plan
- Requirements Traceability Matrix *(expanded)*
- System Design Document
- Program Specifications
- Programming Standards
- Project Plan *(revised)*
- Software Quality Assurance Plan

Outputs:

- Production Platform Acquisition Plan
- Installation Plan *(draft)*
- Software units and modules
- Requirements Traceability Matrix *(expanded)*
- Integration Test Plan *(final)*
- System Test Plan *(final)*
- Project Test File
- Development baselines

□ Transition Plan
□ Operating Documentation *(draft)*
  - Procedure Manual
□ Programmers Reference Manual
□ Administrators Manual
□ Operations Manual
□ Training Plan *(draft)*
□ Project Plan *(revised)*

Key Activities:

Develop Production Platform Acquisition Plan
Develop Production Platform Installation Plan
Establish Programming Environment
Write Programs
Conduct Unit Testing
Establish Development Baselines
Plan Transition to Operational Status
Generate Operating Documentation
Develop Training Program
Revise Project Plan
Conduct Structured Walkthrough(s)
Conduct In-Phase Assessment
Conduct Programming Phase Exit

Forms Utilized

Production Platform Acquisition Plan
Installation Plan
Transition Plan
Training Plan

Methods and Tools

Software:
□ purchase off the shelf software
□ develop custom software
□ decide on platforms (operating systems)
□ user support issues considered

Hardware:
□ hardware systems, vendor selection
□ connectivity allowing for future growth
□ support offered by vendors

Train users:
□ consider users and operators
□ use internal or external training?
□ Training tools
□ manuals
□ video, multimedia presentations
□ tutorials
□ Get feedback on system

Testing of system:
□ ensure quality and reliability
□ use live sample data if possible
□ user acceptance and Test procedures

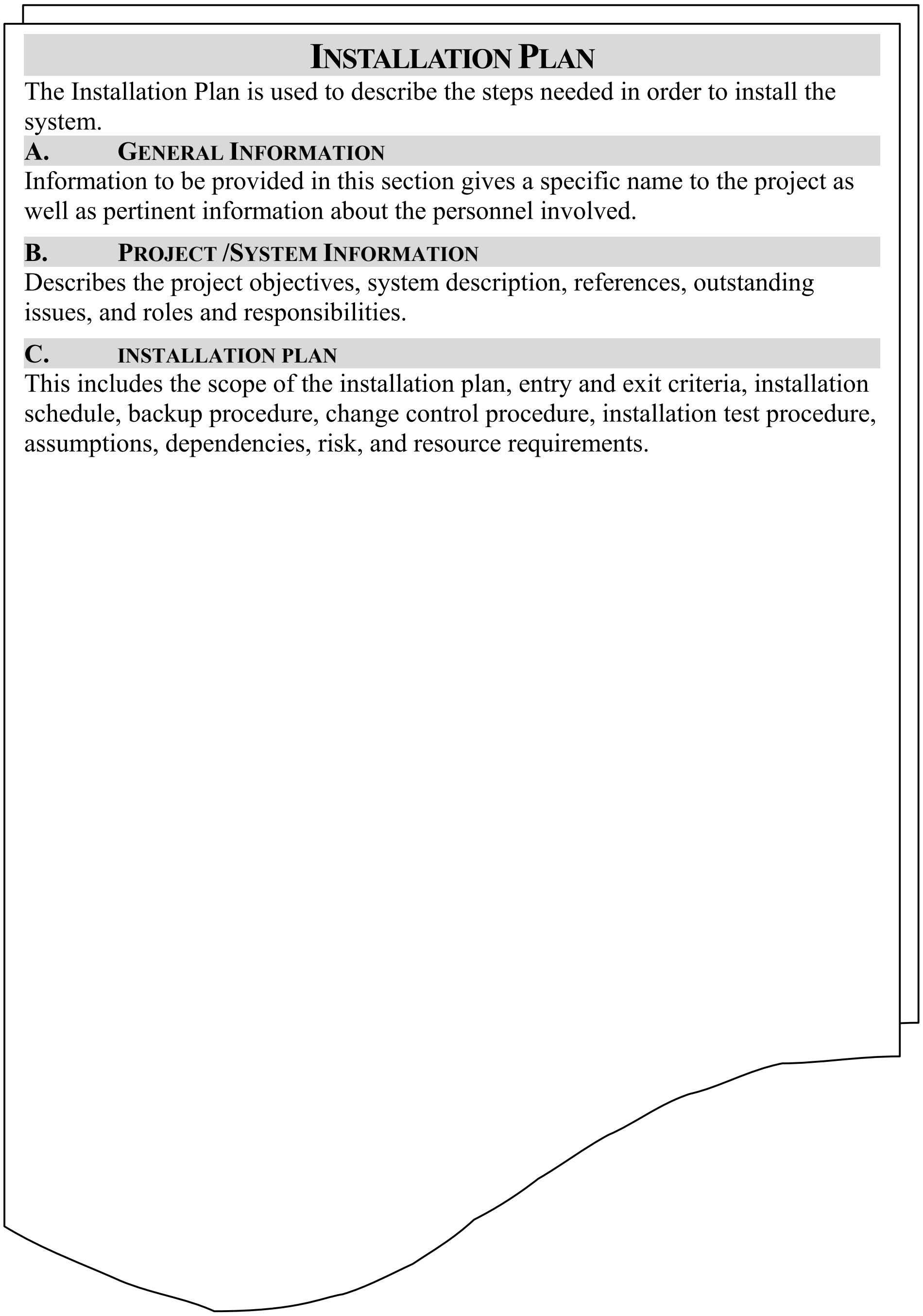
In this phase any hardware or software procured to support the programming effort is installed. Plans are developed for the acquisition and installation of the operating environment hardware and software. A training program is designed and a Training Plan that describes the program is developed.

The activities in this phase result in the transformation of the system design into the first complete representation of the software product. The source code, including suitable comments, is generated using the approved program specifications. If the software product requires a database, any data base utilities are coded. The source code is then grouped into processable units and all high-level language units are compiled into object code. Unit testing is performed to determine if the code satisfies the program specifications and is complete, logical, and error free.

The operating documentation is also developed. The operating documentation is required for installing, operating, and supporting the software product through its lifecycle.

PROJECT MANAGER ROLES AND RESPONSIBILITIES

1) *Procured hardware and/or software is installed;*
2) *Develop plan to acquire and installation of operating environment hardware and software;*
3) *Develop a training plan;*
4) *Source code is generated;*
5) *Database utilities are coded;*
6) *Object code compiled;*
7) *Unit testing performed;*
8) *Operating documentation is developed*



INSTALLATION PLAN

The Installation Plan is used to describe the steps needed in order to install the system.

A. GENERAL INFORMATION

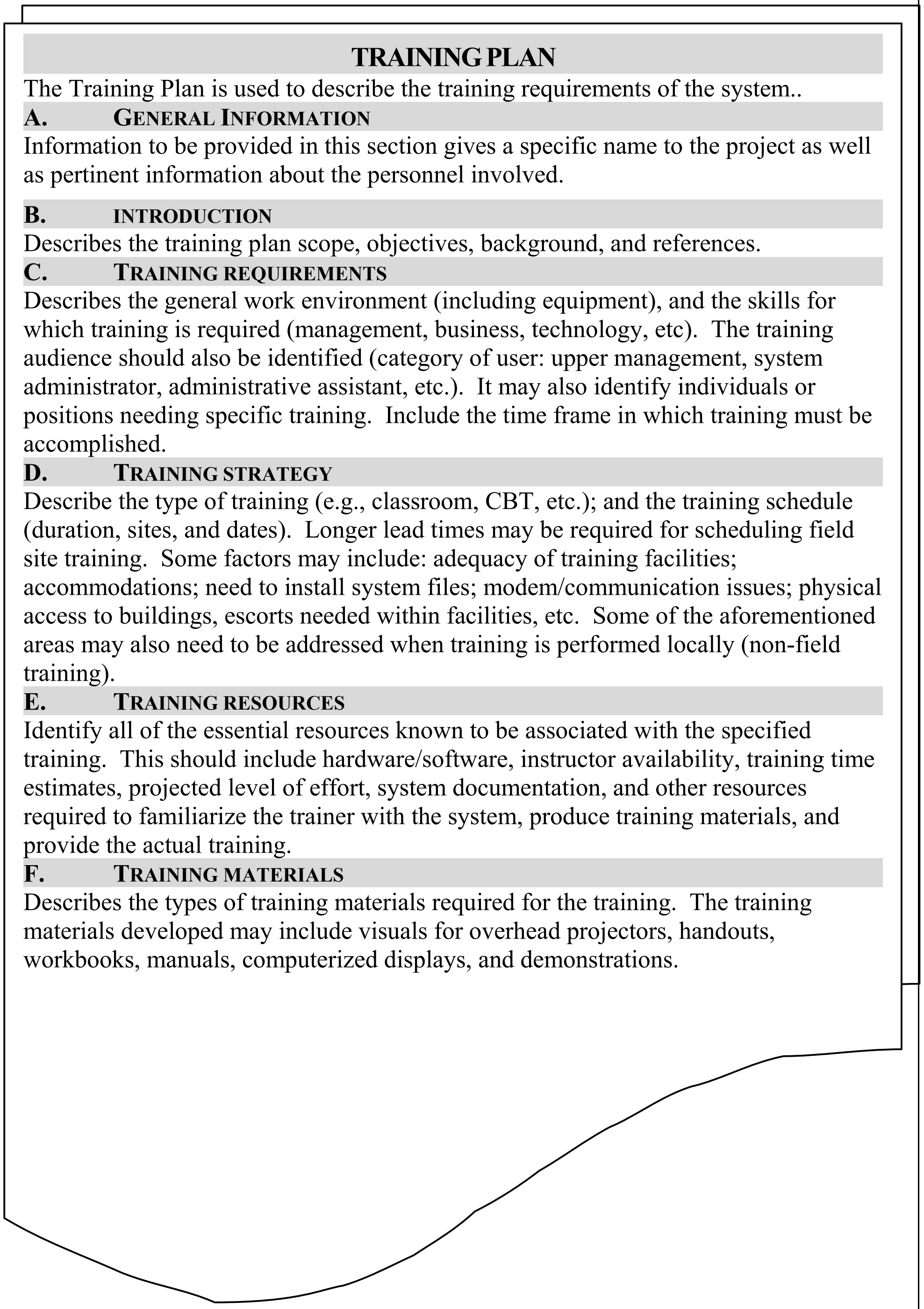
Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. PROJECT /SYSTEM INFORMATION

Describes the project objectives, system description, references, outstanding issues, and roles and responsibilities.

C. INSTALLATION PLAN

This includes the scope of the installation plan, entry and exit criteria, installation schedule, backup procedure, change control procedure, installation test procedure, assumptions, dependencies, risk, and resource requirements.



TRAINING PLAN

The Training Plan is used to describe the training requirements of the system..

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. INTRODUCTION

Describes the training plan scope, objectives, background, and references.

C. TRAINING REQUIREMENTS

Describes the general work environment (including equipment), and the skills for which training is required (management, business, technology, etc). The training audience should also be identified (category of user: upper management, system administrator, administrative assistant, etc.). It may also identify individuals or positions needing specific training. Include the time frame in which training must be accomplished.

D. TRAINING STRATEGY

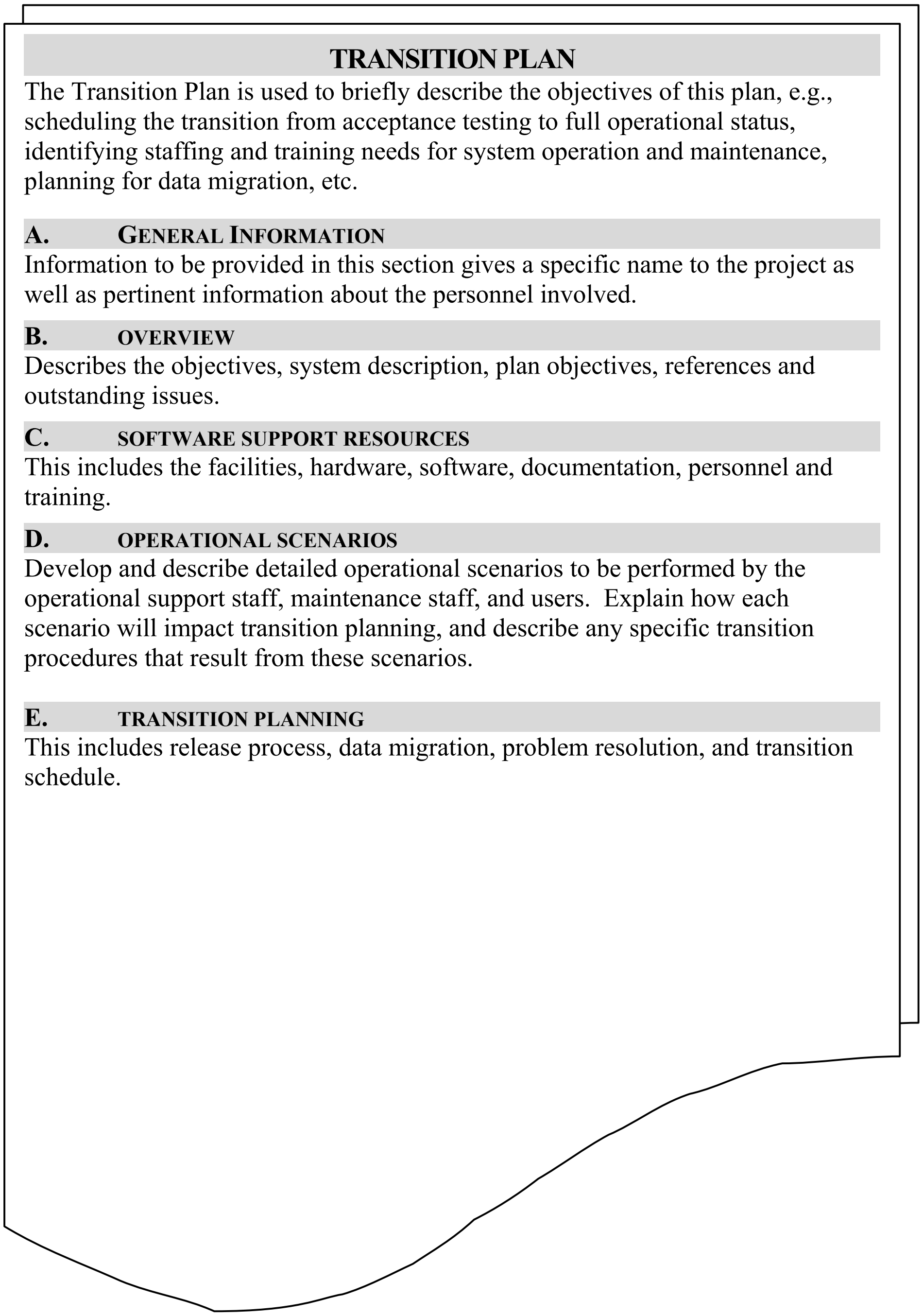
Describe the type of training (e.g., classroom, CBT, etc.); and the training schedule (duration, sites, and dates). Longer lead times may be required for scheduling field site training. Some factors may include: adequacy of training facilities; accommodations; need to install system files; modem/communication issues; physical access to buildings, escorts needed within facilities, etc. Some of the aforementioned areas may also need to be addressed when training is performed locally (non-field training).

E. TRAINING RESOURCES

Identify all of the essential resources known to be associated with the specified training. This should include hardware/software, instructor availability, training time estimates, projected level of effort, system documentation, and other resources required to familiarize the trainer with the system, produce training materials, and provide the actual training.

F. TRAINING MATERIALS

Describes the types of training materials required for the training. The training materials developed may include visuals for overhead projectors, handouts, workbooks, manuals, computerized displays, and demonstrations.



TRANSITION PLAN

The Transition Plan is used to briefly describe the objectives of this plan, e.g., scheduling the transition from acceptance testing to full operational status, identifying staffing and training needs for system operation and maintenance, planning for data migration, etc.

A. General Information

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. OVERVIEW

Describes the objectives, system description, plan objectives, references and outstanding issues.

C. SOFTWARE SUPPORT RESOURCES

This includes the facilities, hardware, software, documentation, personnel and training.

D. OPERATIONAL SCENARIOS

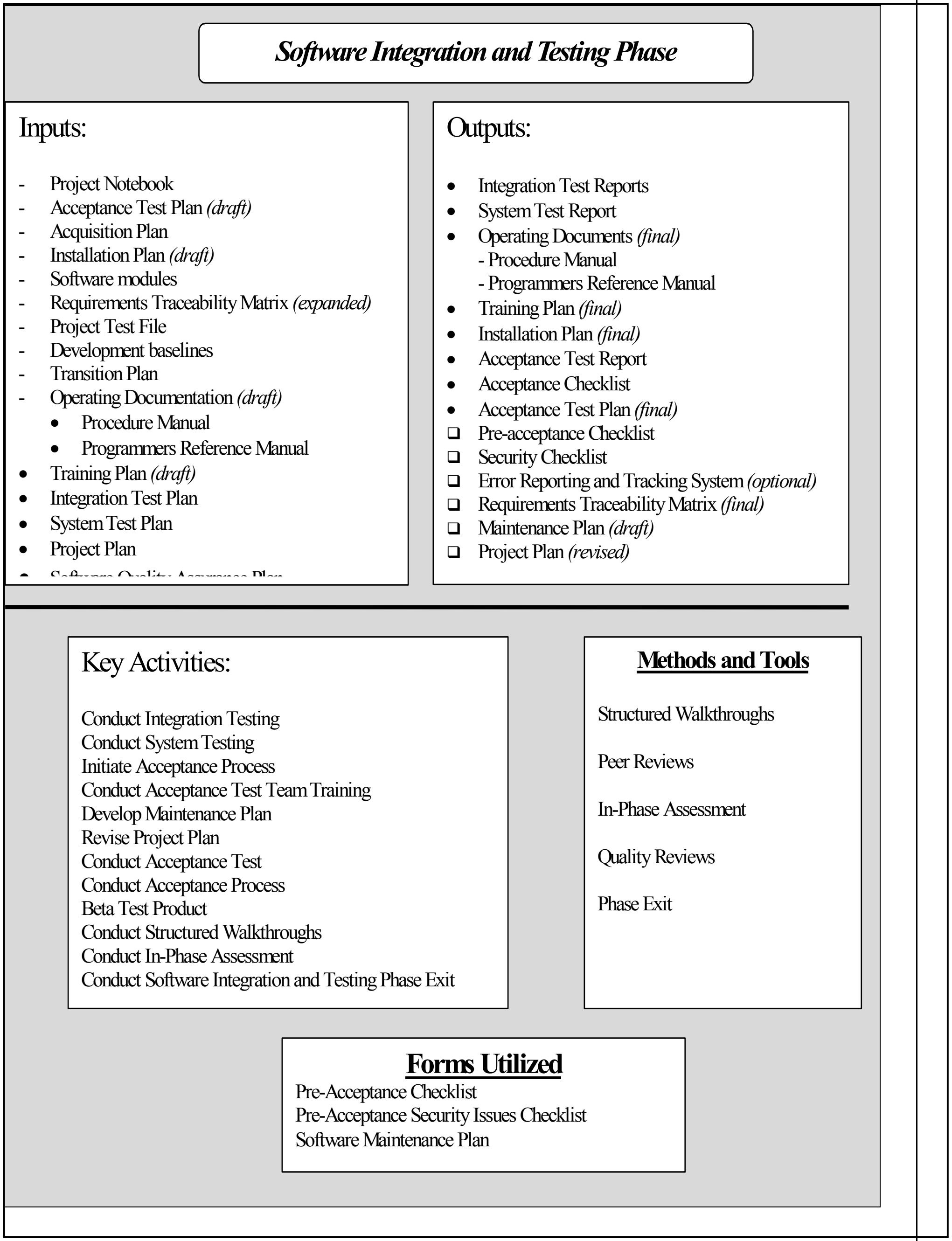
Develop and describe detailed operational scenarios to be performed by the operational support staff, maintenance staff, and users. Explain how each scenario will impact transition planning, and describe any specific transition procedures that result from these scenarios.

E. TRANSITION PLANNING

This includes release process, data migration, problem resolution, and transition schedule.

Notes

Software Integration and Testing Phase



Software Integration and Testing Phase

Inputs:

- Project Notebook
- Acceptance Test Plan *(draft)*
- Acquisition Plan
- Installation Plan *(draft)*
- Software modules
- Requirements Traceability Matrix *(expanded)*
- Project Test File
- Development baselines
- Transition Plan
- Operating Documentation *(draft)*
  - Procedure Manual
  - Programmers Reference Manual
- Training Plan *(draft)*
- Integration Test Plan
- System Test Plan
- Project Plan
- Software Quality Assurance Plan

Outputs:

- Integration Test Reports
- System Test Report
- Operating Documents *(final)*
  - Procedure Manual
  - Programmers Reference Manual
- Training Plan *(final)*
- Installation Plan *(final)*
- Acceptance Test Report
- Acceptance Checklist
- Acceptance Test Plan *(final)*
- □ Pre-acceptance Checklist
- □ Security Checklist
- □ Error Reporting and Tracking System *(optional)*
- □ Requirements Traceability Matrix *(final)*
- □ Maintenance Plan *(draft)*
- □ Project Plan *(revised)*

Key Activities:

Conduct Integration Testing
Conduct System Testing
Initiate Acceptance Process
Conduct Acceptance Test Team Training
Develop Maintenance Plan
Revise Project Plan
Conduct Acceptance Test
Conduct Acceptance Process
Beta Test Product
Conduct Structured Walkthroughs
Conduct In-Phase Assessment
Conduct Software Integration and Testing Phase Exit

Methods and Tools

Structured Walkthroughs

Peer Reviews

In-Phase Assessment

Quality Reviews

Phase Exit

Forms Utilized

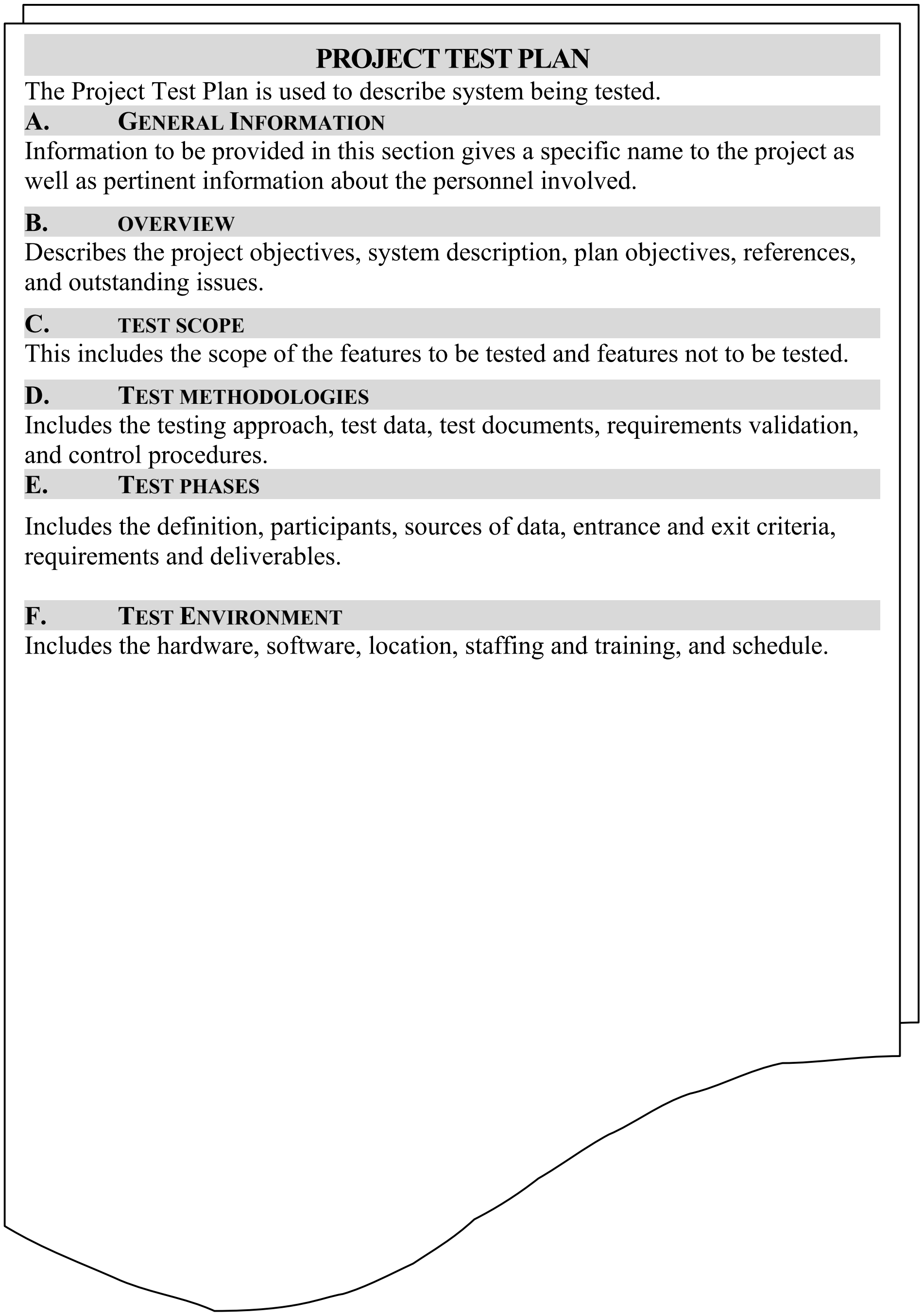
Pre-Acceptance Checklist
Pre-Acceptance Security Issues Checklist
Software Maintenance Plan

Software integration and testing activities focus on interfaces between and among components of the software product, such as functional correctness, system stability, overall system operability, system security, and system performance requirements (e.g., reliability, maintainability, and availability). Software integration and testing performed incrementally provides feedback on quality, errors, and design weaknesses early in the integration process.

In this phase, software components are integrated and tested to determine whether the software product meets predetermined functionality, performance, quality, interface, and security requirements. Once the software product is fully integrated, system testing is conducted to validate that the software product will operate in its intended environment, satisfies all customer requirements, and is supported with complete and accurate operating documentation.

PROJECT MANAGER ROLES AND RESPONSIBILITIES

1) Ensure that software components are integrated and tested;
2) Determine if software product meets predetermined functionality, performance quality, interface, and security requirements;
3) Validate systems testing;
4) Satisfies all customer requirements;
5) Complete and accurate operating documentation



PROJECT TEST PLAN

The Project Test Plan is used to describe system being tested.

A. General Information

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. OVERVIEW

Describes the project objectives, system description, plan objectives, references, and outstanding issues.

C. TEST SCOPE

This includes the scope of the features to be tested and features not to be tested.

D. Test methodologies

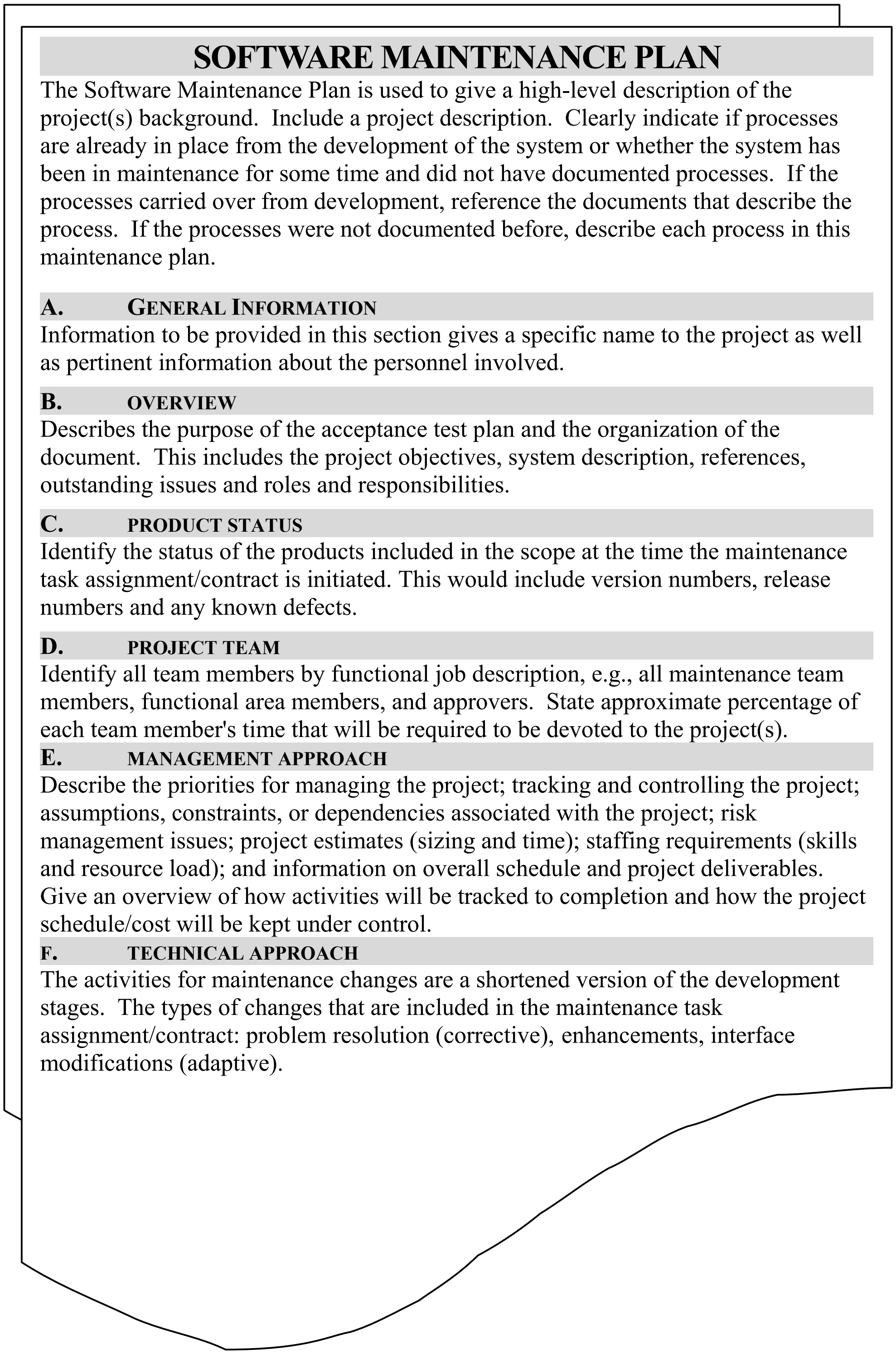
Includes the testing approach, test data, test documents, requirements validation, and control procedures.

E. Test phases

Includes the definition, participants, sources of data, entrance and exit criteria, requirements and deliverables.

F. Test Environment

Includes the hardware, software, location, staffing and training, and schedule.



SOFTWARE MAINTENANCE PLAN

The Software Maintenance Plan is used to give a high-level description of the project(s) background. Include a project description. Clearly indicate if processes are already in place from the development of the system or whether the system has been in maintenance for some time and did not have documented processes. If the processes carried over from development, reference the documents that describe the process. If the processes were not documented before, describe each process in this maintenance plan.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. OVERVIEW

Describes the purpose of the acceptance test plan and the organization of the document. This includes the project objectives, system description, references, outstanding issues and roles and responsibilities.

C. PRODUCT STATUS

Identify the status of the products included in the scope at the time the maintenance task assignment/contract is initiated. This would include version numbers, release numbers and any known defects.

D. PROJECT TEAM

Identify all team members by functional job description, e.g., all maintenance team members, functional area members, and approvers. State approximate percentage of each team member's time that will be required to be devoted to the project(s).

E. MANAGEMENT APPROACH

Describe the priorities for managing the project; tracking and controlling the project; assumptions, constraints, or dependencies associated with the project; risk management issues; project estimates (sizing and time); staffing requirements (skills and resource load); and information on overall schedule and project deliverables. Give an overview of how activities will be tracked to completion and how the project schedule/cost will be kept under control.

F. TECHNICAL APPROACH

The activities for maintenance changes are a shortened version of the development stages. The types of changes that are included in the maintenance task assignment/contract: problem resolution (corrective), enhancements, interface modifications (adaptive).

Notes

Installation and Acceptance Phase

Installation and Acceptance Phase Highlights

Inputs:

- Integration Test Materials
- System Test Materials
- Operating Documents
  - Procedure Manual
  - Programmers Reference Manual
- Training Plan
- Installation Plan
- Conversion Plan
- Acceptance Test Plan
- Pre-acceptance Checklist
- Security Checklist
- Maintenance Plan *(draft)*
- Project Plan *(revised)*
- Software Quality Assurance Plan
- Transition Plan

Outputs:

- Converted data or system files
- Installation Test materials
- Customer Training materials
- Operational software product
- Operating documents
- Maintenance Plan *(final)*
- Project Plan *(revised)*

Key Activities:

Perform Installation Activities
Conduct Installation Tests
Conduct Customer Training
Transition to Operational Status
Revise Maintenance Plan
Revise Project Plan
Conduct Structured Walkthrough(s)
Conduct In-Phase Assessment
Conduct Installation and Acceptance Phase Exit

Methods and Tools

Complete documentation for:

- ❑ computer operators
- ❑ data entry personnel/end users
- ❑ maintenance programmers
- ❑ management

Install system:

- ❑ one of four approaches may be used
- ❑ log all errors/problems to aid in maintenance

Installation and acceptance of the software product are initiated after the system test has been successfully completed. This phase involves the activities required to install the software, databases, or data that comprise the software product onto the hardware platform at the site(s) of operation. The objectives of the activities in this phase are to verify that the software product meets design requirements and to obtain the system owner's acceptance and approval of the software product. The activities associated with this phase should be performed each time the software product is installed at an acceptance test site or production site.

Customer training may be required to complete the installation process. A description of the training necessary for programmers, testers, customers, and operations staff is provided in the Training Plan.

PROJECT MANAGER ROLES AND RESPONSIBILITIES

1) *Installation and acceptance of software product;*
2) *Install software and databases onto hardware;*
3) *Verify that software meets design;*
4) *Obtain acceptance and approval;*
5) *Customer training*

Notes

Emergency Maintenance

This section describes an iterative process for conducting software emergency maintenance activities. The process prescribes a minimal set of criteria that are necessary for project management and quality assurance processes; control; and management of the planning, execution, and documentation of emergency maintenance activities. The use of automated tools to facilitate requirements definition, design, coding, and system documentation is encouraged. The selection and implementation of tools varies among sites and organizations.

The basic maintenance process model includes input, process, output, and control for emergency maintenance. It is based on the same software development principles and preferred practices that lower risk and improve quality during the planning and development phases of the lifecycle. The process model supports the concept that planned software changes should be grouped and packaged into scheduled releases that can be managed as projects. This proven approach allows the maintenance team to better plan, optimize use of resources, take advantage of economies of scale, and better control outcome in terms of both schedule and product quality.

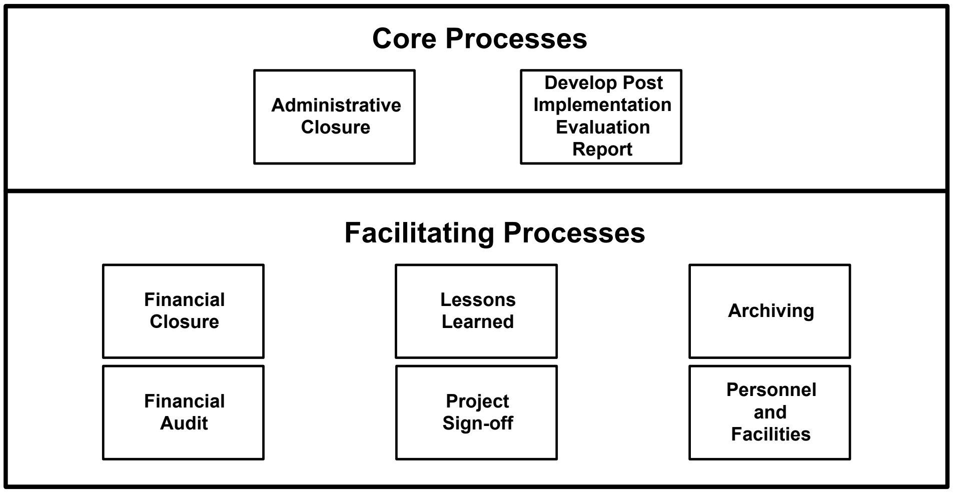
Each organization performing emergency maintenance activities should have a local documented procedure for handling emergency changes that cannot be implemented as part of a scheduled release. Generally, these changes include fixes to correct defects and updates to meet unscheduled business or legal requirements. Emergency changes should be integrated into the next release for full regression testing and documentation updates.

PROJECT MANAGER ROLES AND RESPONSIBILITIES

1)
2)
3)
4)
5)

Notes

Project Closeout Phase



Core Processes

Administrative Closure

Develop Post Implementation Evaluation Report

Facilitating Processes

Financial Closure

Lessons Learned

Archiving

Financial Audit

Project Sign-off

Personnel and Facilities

🗐 = Deliverable ⌘ = Process

The Project Closeout Phase involves the administrative and financial efforts needed to close out a project after the work has been completed. Also during the Closeout Phase, the product is transferred to the customer.

PROJECT MANAGER ROLES AND RESPONSIBILITIES

- Obtain customer and management sign-off.
- Close out any open action items.
- Develop Post Implementation Evaluation Report.
- Conduct lessons learned session.
- Close out any financial accounts or charge codes.
- Archive all project data.
- Assist as needed with any post-project delivery audits.
- Assist Contract Administrator(s) in contract closeout.
- Assist in resource transition from the project.
- Celebrate success.

CLOSEOUT PHASE CORE PROCESSES

⌘ Administrative Closure – Administrative Closure is the process of preparing closure documentation of the project deliverables for the customer as well as taking other administrative actions to ensure that the project and its assets are redistributed.

🗐 Post Implementation Evaluation Report (PIER) – A Post Implementation Evaluation Report documents the successes and failures of the project. It provides a historical record of the planned and actual budget and schedule. Other selected metrics on the project can also be collected, based upon agency organizational procedures. The report also contains recommendations for other projects of similar size and scope. A sample of the format of the PIER is provided on page 57.

CLOSEOUT PHASE FACILITATING PROCESSES

⌘ Financial Closure – Financial Closure is the process of completing and terminating the financial and budgetary aspects of the project being performed. It includes both (external) contract closure and (internal) project account closure.

⌘ Lessons Learned – A Lessons Learned session is a valuable closure and release mechanism for team members, regardless of the project's success. It allows for an exchange of observations of the project's performance to aid with future projects of a similar nature. Lessons Learned should be sent to the Agency Project Management Office (and possibly the State Office of Project Management at DMB) for input into other projects.

⌘ Archiving – Archiving is creating and storing a hard and/or soft copy of all documents related to the project. The summary project management information collected includes information such as a description of the project, a project organization chart, budgeted and actual cost, and budgeted and actual schedule.

⌘ Financial Audit – A Financial Audit is a thorough examination of a project, by an evaluation team including a detailed overview of the project's financial procedures, budgets, records, etc. It may deal with a project as a whole or the separate individual parts of a project.

⌘ Project Sign-off –A final meeting with all necessary stakeholders may be held to review the product delivered against the baseline requirements and specifications. The final deliverable of this meeting should be a statement created by the project manager that describes the project's final deliverables in comparison with the authorized project baseline documents. This sign-off is contained as part of the PIER documentation.

⌘ Personnel and Facilities – This closure process is the reassignment and reallocation of agency personnel and equipment that have been used during the project.

CLOSEOUT PHASE DELIVERABLES

🗐 Post Implementation Evaluation Report – This report records and appraises actions conducted on the project**.** The report contains Lessons Learned, which provide an opportunity to better understand project successes and shortcomings, which can be effectively applied to the next project.

SYSTEM DEVELOPMENT LIFE CYCLE COMPONENTS OF THE CLOSEOUT PHASE

- ⌘ Maintenance – Once the product(s) of the Information Technology project is created and implemented, it goes into maintenance mode in which it will stay throughout its useable life, or until it is retired or replaced. In order to support the maintenance effort, all customer manuals and other useful documentation should be turned over to the system operators and customers.
- ⌘ Service Level Agreement – This agreement process, and document, indicates that the project team has taken the steps to procure necessary protracted services and maintenance for the product.

CLOSEOUT PHASE – OTHER IMPORTANT INFORMATION

- ⌘ Closing out a project is not as easy as one would think. There are several small details that must be taken care of such as account closure and personnel reassignment. Make a checklist that pertains to your project of things to closeout, and follow up on each item.
- ⌘ Procedures for future changes may also be of concern at project closeout. Accordingly, establish any necessary actions to accommodate product changes, formally or informally, in the future.
- ⌘ Don't forget to celebrate your project's success. Award those who supported the project and let the team members know that their participation was appreciated.

POST IMPLEMENTATION EVALUATION REPORT

The purpose of the Post Implementation Evaluation Report (PIER) is to document the successes and failures of the project and to provide a historical record of the planned and actual budget and schedule.

A. GENERAL INFORMATION

Information to be provided in this section gives a specific name to the project as well as pertinent information about the personnel involved.

B. STAFFING AND SKILLS

Describe how the staffing and skill needs for this project were determined and managed. Describe changes to these needs during the project.

C. PROJECT ORGANIZATIONAL STRUCTURE

Provide an organization chart that was used for the project. Describe any changes made to the structure along the way and why.

D. SCHEDULE MANAGEMENT

Provide the baseline project schedules and final project schedules (as attachments if necessary). Describe the process used for controlling schedules as well as actions taken to correct any problems.

E. COST MANAGEMENT

Describe cost and budget results of the project in comparison to the baseline.

F. RISK MANAGEMENT

Provide a description of the risks identified and how they were handled.

G. QUALITY CONTROL

Describe how quality control was involved in this project.

H. CONFIGURATION MANAGEMENT

Describe how the configuration management process was utilized.

- continued -

POST IMPLEMENTATION EVALUATION REPORT - CONTINUED

I. COMMUNICATIONS MANAGEMENT

Describe the project communication process, its effectiveness and any changes made to the communications plan during the project.

J. CUSTOMER EXPECTATIONS MANAGEMENT

Describe how customer expectations were managed. Were expectations clear from the beginning? How were expectations different than expected?

K. LESSONS LEARNED

Describe the successes and shortcomings of the project activities.

L. PROJECT SIGN-OFF

Delineates the functional areas of the project team have taken all steps to present deliverables and project activities are closed out.

Notes

APPENDICES

GLOSSARY

The following is a list of common terms and acronyms used within the Systems Development Lifecycle and industry. If you need further information on any of the subjects in the following list, please consult the State of Michigan's Systems Development Lifecycle and the variety of sources listed in the Resources and Reference list within the Appendix of the Systems Development Lifecycle.

- A –

Acceptance criteria
The criteria that a software component, product, or system must satisfy in order to be accepted by the system owner or other authorized acceptance authority.

Acceptance process
The process used to verify that a new or modified software product is fully operational and meets the system owner's requirements. Successful completion of the acceptance process results in the formal transfer of the software product responsibilities from development to maintenance personnel.

Acceptance testing
Formal testing conducted to determine whether or not a software product or system satisfies its acceptance criteria and to enable the system owner to determine whether or not to accept the product or system.

Activity
A major unit of work to be completed in achieving the objectives of a software project. An activity incorporates a set of tasks to be completed, consumes resources, and results in deliverables. An activity may contain other activities in a hierarchical manner. All project activities should be described in the Project Plan.

Algorithm
A finite set of well-defined rules for the solution to a problem in a finite number of steps. Any sequence of operations for performing a specific task.

Allocated requirements
The subset of the system requirements that are to be implemented in the software components of the system.

Anomaly
Anything observed in the operation or documentation of software that deviates from expectations based on previously verified software products or documents.

Application
Software products designed to fulfill specific needs.

Assumption
A condition that is taken to be true without proof or demonstration.

Audit
An independent examination of a deliverable to assess compliance with specifications, standards, quality or security requirements, contractual agreements, or other predetermined criteria.

- B -

Baseline
A set of configuration items (software components and documents) that has been formally reviewed and agreed upon, that serves as the basis for further development, and that can be changed only through formal change control procedures.

Baselined requirements
The set of project requirements that have been approved and signed off by the system owner during the Requirements Definition Phase. The software product design is based on these requirements. The baselined requirements are placed under configuration control.

- C -

Code
Computer instructions and data definitions expressed in a programming language or in a form that is output by an assembler, compiler, or other translator.

Code generator
A software tool that accepts as input the requirements or design for a computer program and produces source code that implements the requirements or design.

Code review
A meeting at which software code is presented to project personnel, managers, users, or other functional areas for review, comment, or approval.

Component
One of the parts that make up a system. A component may be hardware, software, or firmware and may be subdivided into other components.

Computer-Aided Software Engineering (CASE)
The use of computers to aid in the software engineering process may include the application of software tools for software design, requirements tracing, code production, testing, document generation, and other software engineering activities.

Configuration control
An element of configuration management consisting of the evaluation, coordination, approval/disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.

Configuration control board
A group of people responsible for evaluating and approving/disapproving proposed changes to configuration items, and for ensuring implementation of approved changes.

Configuration item
An aggregate of hardware or software components that are designated for configuration management and treated as a single entity in the configuration management process.

Configuration management
See Software configuration management

Constraint
A restriction, limit, or regulation that limits a given course of action or inaction.

Cost estimate
A formal estimate of the cost to develop and support a project. Estimates should reflect all activities such as design, development, coding, distribution, service, and support of the product; staffing; training and travel expenses; subcontractor activities; contingencies; and cost for external services (e.g., technical documentation production and Quality Assurance audits and reviews).

- D -

Deliverable
Any tangible item that results from a project function, activity, or task. Examples of deliverables include process descriptions, plans, procedures, computer programs, and associated documentation, which may or may not be intended for delivery to the system owner and other project stakeholders. A deliverable that is identified in the Project Plan and is formally delivered to the system owner and other project stakeholders for review and approval.

Dependency
A relationship of one task to another where the start or end date of the second task is related to the start or end date of the first task.

Design
The process of defining the architecture, components, interfaces, and other characteristics of a software product or component.

Design specification
A document that describes the design of a software component, product, or system. Typical contents include architecture, control logic, data structures, input/output formats, interface descriptions, and algorithms.

- F -

Feasibility
The degree to which the requirements, design, or plans for a software product or system can be implemented under existing constraints.

Functional area
Any formally organized group involved in the development and maintenance of software or the support of development and maintenance efforts, or other group whose input is required to successfully implement software project. Examples of functional areas include software engineering services, technical writing, quality assurance, security, and telecommunications.

Functional design phase
The period of time in the software lifecycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy project requirements.

Functional requirement
A requirement that specifies a function that a software component, product, or system must be able to perform.

Functional test plan
A plan for testing each function across one or more units. The plan describes how the functional testing occurs and the test procedure/test cases that will be used. The plan includes procedures for creating the test environment that allows all functions to be executed; the entry and exit criteria for starting and ending the function-testing period; and the schedule followed for starting and ending each test.

Functional test procedures
Procedures for each function or combination of functions to be tested. Procedures fully describe how the function is tested. Expected output from each test procedure is identified to compare the planned output to actual output.

Functional testing
Testing conducted to evaluate the compliance of a software product with specified functional requirements. Testing that focuses on the outputs generated in response to selected inputs and execution conditions.

- H -

Hardware
Physical computer and other equipment used to process, store, or transmit computer programs or data.

Hierarchy
A structure in which components are ranked into levels of subordination.

- I -

Implementation requirements
A requirement that supports the development and maintenance concepts and approaches in the areas of operating environment, conversion, installation, training, and documentation.

Incremental development
A software development technique in which requirements definition, design, implementation, and testing occur in an overlapping, iterative (rather than sequential) manner, resulting in incremental completion of the overall software product.

Information engineering
A development methodology where models are created to improve the users' ability to understand and define the functions and flow of information within their organization. A business model is developed to identify the key areas of interest for the business, the tasks required for each area, and the activities that make up each task. The business model prioritizes and identifies top management goals and then establishes the information needs necessary to reach those goals. A data model is developed to describe the data and the relationships among data. The data model further divides the business model into user-defined relationships (e.g., entity relationship model).

Inspection
A static analysis technique that relies on visual examination of development products to detect errors, violations of development standards, and other problems. Code inspection and design inspection are two types.

Integration testing
An orderly progression of testing in which software components are combined and tested to evaluate the interaction between them.

Integrity
The degree to which a software component, product, or system prevents unauthorized access to, or modification of, computer programs or data.

Interactive analysis and design
A development methodology that uses facilitated team techniques, such as Joint Application Development or Rapid Application Development, to rapidly develop project requirements that reflect the users' needs in terminology that the users understand. Group facilitation techniques are especially important when several user organizations have unique project requirements that are specific to their mission and goals.

Interface requirement
A requirement that specifies an external item with which a software product or system must interact, or that sets forth constraints on formats, timing, or other factors caused by such an interaction.

Interface testing
Testing conducted to evaluate whether software components pass data and control correctly to one another.

Interview technique
A technique for the identification, analysis, and documentation of the project requirements. The project team conducts a series of interviews with users to identify the users' perceived automated functional needs, analyzes the information gathered during the interviews, and develops the requirements.

- K -

Key process area
Software engineering processes identified by the Software Engineering Institute Capability Maturity Model where a project team should focus its efforts to achieve consistently high quality software products.

- L -

Lifecycle See Software lifecycle.

- M -

Maintenance
The process of supporting a software product or system after delivery to maintain operational status, correct faults, improve performance or other attributes, or adapt to a changed environment.

Menu-driven
Pertaining to a system or mode of operation in which the users direct the software through menu selections.

Methodology
A collection of methods, procedures, and standards that defines an integrated synthesis of engineering approaches to the development of a deliverable.

Milestone
A scheduled event for which an individual or team is accountable and that is used to measure progress.

Module
A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading. A logically separable part of a program.

Module testing
Testing of individual software modules or groups of related modules to verify the implementation of the design.

- O -

Organization
An organization is a unit with a company or other entity within which projects are managed as a whole. An operational definition for “organization” is the scope of an appraisal or process improvement effort. Organization analysis is necessary to define exactly what the scope will be. All projects within an organization share a common top-level manager and common policies.

- P -

Performance requirement
A requirement that imposes conditions on a functional requirement (e.g., a requirement that specifies the speed, accuracy, or memory usage with which a given function must be performed).

Phase
A partition of the software lifecycle that reduces a project to manageable size and represents a meaningful and measurable set of related tasks that are performed to obtain specific deliverables.

Planning phase
The initial phase in the software lifecycle during which the system owner/users' needs and expectations are identified, the feasibility of the project is determined, and the Project Plan is developed.

Platform
A specific computer and operating system on which a software product is developed or operated.

Portability
The ease with which a software component, product, or system can be transferred from one hardware or software environment to another.

Procedure
A written description of a course of action to be taken to perform a given task.

Process
An ordered set of steps performed for a given purpose. Processes define or control the development of the project deliverables. The use of processes will ensure a consistent methodology across all platforms in producing the lifecycle deliverables.

Programmer's reference manual
A deliverable that provides information necessary to maintain or modify software for a given computer system. Typically described are the equipment configuration, operational characteristics, programming features, input/output features, and compilation or assembly features of the computer system.

Programming phase
The period of time in the software lifecycle during which a software product is created from the design specifications and testing is performed on the individual software units.

Project
An undertaking requiring concerted effort that is focused on developing or maintaining a specific software product or system. A project has its own funding, cost accounting, and delivery schedule.

Project lifecycle
The software lifecycle selected for the project and approved by the system owner and other project stakeholder(s).

Project manager
The individual with total business responsibility for all software activities of a project. The project manager directs controls, administers, and regulates a project.

Project notebook
A central repository of material pertinent to a project. Contents typically include all deliverables, memos, plans, technical reports, and related items.

Project plan
A document that describes the technical and management approach to be followed for a project. The plan typically describes the work to be done, the resources required, the methods to be used, the procedures to be followed, the schedules to be met, and the way the project will be organized. The plan

includes a list of deliverables, actions required, and other key events needed to accomplish the project.

Project team
The project manager, analysts, programmers, and other staff assigned as the core group for a project. The project team may include representatives of the other functional areas (e.g., technical writer and telecommunications expert) responsible for contributing to the development, installation, and maintenance of the software product.

Project Test Plan
Defines all test activities required to assure that the software product will perform satisfactorily for all users. As a minimum, the plan should include descriptions for unit testing, integration testing, system testing, and acceptance testing.

Prototyping
A technique for developing and testing a preliminary version of the software product (either as a whole or in modular units) in order to emulate functionality without such encumbering features as error handling, help messages, security controls, and other utilities that are not part of the design logic. This allows the

project team to test the overall logic and workability of required functions and provides a model by which the project team and users can jointly determine if the software requirements meet the intended objectives. Prototyping is often used in conjunction with interactive analysis and design techniques.

Pseudocode
A combination of programming language constructs and natural language used to express a computer program design.

- R -

Rapid prototyping
A type of prototyping in which emphasis is placed on developing prototypes earlier in the development process to permit early feedback and analysis in support of the development process.

Reference
A document(s) or other material that is useful in understanding more about an activity.

Regression testing
Selective retesting of a software component to verify that modifications have not caused unintended effects and that the software component still complies with its specified requirements.

Reliability
The ability of a software component to perform its required functions under stated conditions for a specified period of time.

Requirement
A condition or capability needed by a system owner/user to solve a problem or achieve an objective. A condition or capability that must be met or possessed by the software product to satisfy a contract, standard, specification, or other formally imposed documents.

Requirements analysis
The process of studying system owner/user(s) needs to arrive at a definition of system, hardware, or software requirements.

Requirements definition phase
The period of time in the software lifecycle during which the requirements for a software product are defined and documented.

Requirements management
A Software Engineering Institute Capability Maturity Model key process area designed to establish a common understanding between the system owner/user and the project team regarding the system owner/users' software requirements. This understanding forms the basis for estimating, planning, performing, and tracking the project's activities throughout the lifecycle.

Requirements specification
A deliverable that specifies the manual and automated requirements for a software product in non-technical language that the system owner/users can understand. Typically included are functional requirements, performance requirements, and interface requirements. Describes in detail what will be delivered in the product release.

Retirement
Permanent removal of a system or software product from its operational environment.

Reusability
The degree to which a software module or other deliverable can be used in more than one computer program or software system.

Reverse engineering
A development methodology in which the software developments process is performed in reverse. The technique involves the examination of an existing software product that has characteristics that are similar to the desired product. Using the existing code as a guide, the requirements for the product are defined,
analyzed, and abstracted all the way back to specifications. Any required code changes can be made based on a specification-like format. Ideally, the specifications would be edited and passed to a code generator that would trigger automatic documentation and revisions. Once testing is complete, the revised code is placed into production.

Risk
The possibility of suffering loss.

Risk management
An approach to problem analysis that is used to identify, analyze, prioritize, and control risks.

- S -

Software
Computer programs, procedures, and associated documentation and data pertaining to the operation of a software product or system.

Software configuration item
An aggregation of software that is designated for configuration management and treated as a single entity in the configuration management process.

Software configuration management
(1) A discipline that effectively controls and manages all modifications to a software component, product, or system. Technical and administrative processes and tools are used to identify and document the functional and physical characteristics of the configuration items, manage and track changes to those
items, record and report change processing and implementation status, and verify compliance with specified requirements. (2) A Software Engineering Institute Capability Maturity Model key process area designed to establish and maintain the integrity of the software deliverables throughout the project's lifecycle.

Software development plan
See Software Project Management Plan.

Software lifecycle
The period of time that begins when a software product is conceived and ends when the software is retired. A network of phases and processes that function together to guide the development and maintenance of software products. Each process produces a set of deliverables as it moves through the lifecycle.

Software procurement
The process by which software is acquired. The process is defined by the procurement plan, which identifies business and/or technical requirements development, purchase methodology, budget, resources for a Joint Evaluation Committee (executive and core team), high-level testing plan, required approvals, and contracting methods. The software may be an off-the-shelf package that is a project product, or it may be testing, QA, operating system, database management, ...software that supports the project process.

Software project planning
A Software Engineering Institute Capability Maturity Model key process area designed to establish reasonable plans for performing software engineering and for managing the software project.

Software project management plan
The controlling document for managing a software project. The plan defines the technical and managerial functions, activities, and task necessary to satisfy the requirements of a software project, as defined in the project agreement. Synonymous with software development plan.

Software project tracking and oversight
A Software Engineering Institute Capability Maturity Model key process area designed to provide adequate visibility into actual project progress so that management can take effective actions when the project's performance deviates significantly from the plans.

Software quality assurance
A Software Engineering Institute Capability Maturity Model key process area designed to provide management with appropriate visibility into the software engineering processes being used by the project team and the deliverables being built.

Software system
A software product and the documentation, hardware, and communications needed to implement and operate the product and accomplish a specific function or set of functions.

Specification
A document that specifies in a complete, precise, verifiable manner the requirements, design, behavior, or other characteristics of a software component, product, or system.

Stakeholder
The Agency individual(s) with decision-making authority over a project or group of projects.

Standard
Mandatory requirements employed and enforced to prescribe a disciplined, uniform approach to software development and maintenance.

Structured analysis
An analysis technique that uses a graphical language to build models of software products or systems. The four basic features in structured analysis are data flow diagrams, data dictionaries, procedure logic representations, and data store structuring techniques.

System
A collection of hardware, software, firmware, and documentation components organized to accomplish a specific function or set of functions.

System design document
A deliverable that describes the solution to the automation task as described by the requirements. Contains sufficient detail to provide necessary direction for writing the Program Specifications and allows developers maximum technical freedom.

System design phase
A phase in the lifecycle model during which the designs for the software product architecture, software components, interfaces, and data are refined and expanded to the extent that the design is sufficiently complete to be implemented.

System owner
The organizational unit that funds and has approval authority for the project. Typically, system owners are also system users.

Systems development lifecycle
The State of Michigan's methodology that identifies the processes, activities, tasks, management responsibilities, and deliverables that are required for each software development and maintenance project. Deviations from the methodology are managed on a project-by-project basis. A key objective of the methodology is to provide measurable, repeatable processes to assure that project development and maintenance methodologies are consistent throughout the agency information technology environment.

System testing
Testing conducted on a complete, integrated software product or system to evaluate compliance with its specified requirements.

- T -

Task
The smallest unit of work subject to management accountability. A task is a well-defined work assignment for one or more project team members. Related tasks are usually grouped to form activities. A task is the lowest level of work division typically included in the Project Plan and Work Breakdown Structure.

Telecommunications
The science and technology of communications by electronic transmission of impulses, as by telephone or e-mail.

Test bed
An environment containing the hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test.

Test case
A set of test inputs, execution conditions, and expected results that are developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Test criteria
The criteria that a software component or product must meet in order to pass a given test.

Test design
Documentation specifying the details of the test approach for a software feature or combination of software features and identifying the associated tests.

Test documentation
Documentation describing plans for, or results of, the testing of a software component or product. Documents typically include test case specifications, test incident reports, test logs, test plans, test procedures, and test reports.

Test item
A software item that is the object of testing.

Test log
A chronological record of all relevant details about the execution and results of a test.

Test phase
The period of time in the software lifecycle in which the components of a software product are evaluated and integrated, and the software product is evaluated to determine whether or not the requirements have been satisfied.

Test plan
A document specifying the scope, approach, resources, and schedule of intended testing activities. The plan identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.

Test procedure
Detailed instructions for the setup, execution, and evaluation of the results for a given test case.

Test report
A document that describes the conduct and results of the testing carried out for a software component or product.

Testing
An activity in which a software component or product is executed under specified conditions, the results are observed and recorded, and an evaluation is made.

Traceability
The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor relationship to one another.

Transaction analysis
A technique used to derive structured charts for a software product that will process transactions. Transaction analysis is used to divide complex data flow diagrams into smaller, simpler data flow diagrams--one for each transaction that the product or system will process. Structure charts are developed from the simple data flow diagrams. The individual structure charts for the separate
transactions are then combined to form one large structure chart that is very flexible and can accommodate user changes.

Transition plan
The system is transitioned into operational status. The transition plan contains materials, operating documents, and other pertinent records. These materials are turned over to the maintenance staff.

- U -

Unit
A separately testable element specified in the design of a computer software component. A software component that is not subdivided into other components.

Unit testing
Testing of individual hardware or software units or groups of related units. The isolated testing of each flowpath of code with each unit. The expected output from the execution of the flowpath should be identified to allow comparisons of the planned output against the actual output.

Usability
The ease with which a user can learn to operate, prepares inputs for, and interprets outputs of a software product.

User
The general population of individuals who use a software product or system. User activities can include data entry; read only; add, change and delete capabilities; querying; and report generation.

User interface
An interface that enables information to be passed between a user and hardware or software components of a computer system.

User manual
A document that presents the information necessary to use a software product to obtain desired results. Typically described are products or component capabilities, limitations, options, permitted inputs, expected outputs, possible error messages, and special instructions.

- V -

Validation
The process of evaluating software at the end of the software development process to assure compliance with established software and system requirements.

Verification
The process of evaluating a software product to determine whether or not the deliverables of a phase of the software lifecycle fulfill the requirements established during the previous phase.

- W -

Walkthrough
An analysis technique in which a team of subject matter experts review a segment of code or documentation, ask questions, and make comments about possible errors, violation of development standards, and other problems.

Capability Maturity Model Overview

The Capability Maturity Model (CMM) comes from the Software Engineering Institute (SEI) of Carnegie Mellon University. The SEI has conducted significant research into several areas of business process improvement and reengineering over the past several years.

The CMM, described briefly below, is a professionally recognized model for process development within software-based organizations. The ultimate intent of applying this model for software development in order for State of Michigan agencies to strive to a level five maturity in Project Management. If successful, this practice will eventually spread to other lifecycle areas and throughout Michigan State Government.

CMM Defined

The Capability Maturity Model for Software describes the principles and practices underlying software process maturity and is intended to help software organizations improve the maturity of their software processes in terms of an evolutionary path from ad hoc, chaotic processes to mature, disciplined software processes. The CMM is organized into five maturity levels:

1) Initial. The software processes are characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort and heroics.

2) Repeatable. Basic management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.

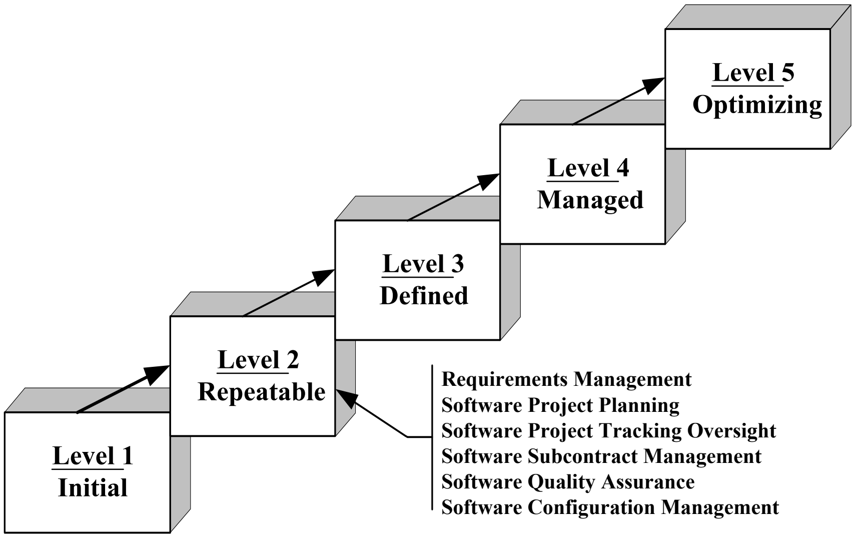
3) Defined. The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.

4) Managed. Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.

5) Optimizing. Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

Predictability, effectiveness, and control of an organization's software processes are believed to improve as the organization moves up these five levels. While not rigorous, the empirical evidence to date supports this belief.

CAPABILITY MATURITY MODEL PROCESS LEVELS


Level 5
Optimizing
Level 4
Managed
Level 3
Defined
Level 2
Repeatable
Level 1
Initial
Requirements Management
Software Project Planning
Software Project Tracking Oversight
Software Subcontract Management
Software Quality Assurance
Software Configuration Management

CMM Process Decomposition

Except for Level 1, each maturity level is decomposed into several Key Process Areas (KPA) that indicate the areas an organization should focus on to improve its software process.

The Key Process Areas at Level 2 focus on the software project's concerns related to establishing basic project management controls. They are Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance, and Software Configuration Management.

The Key Process Areas at Level 3 address both project and organizational issues, as the organization establishes an infrastructure that institutionalizes effective software engineering and management processes across all projects. They are Organization Process Focus, Organization Process Definition, Training Program, Integrated Software Management, Software Product Engineering, Intergroup Coordination, and Peer Reviews.

The Key Process Areas at Level 4 focus on establishing a quantitative understanding of both the software process and the software work products being built. They are Quantitative Process Management and Software Quality Management.

The Key Process Areas at Level 5 cover the issues that both the organization and the projects must address to implement continual, measurable software process improvement. They are Defect Prevention, Technology Change Management, and Process Change Management.

Each Key Process Area is described in terms of the key practices that contribute to satisfying its goals. The key practices describe the infrastructure and activities that contribute most to the effective implementation and institutionalization of the key process area.

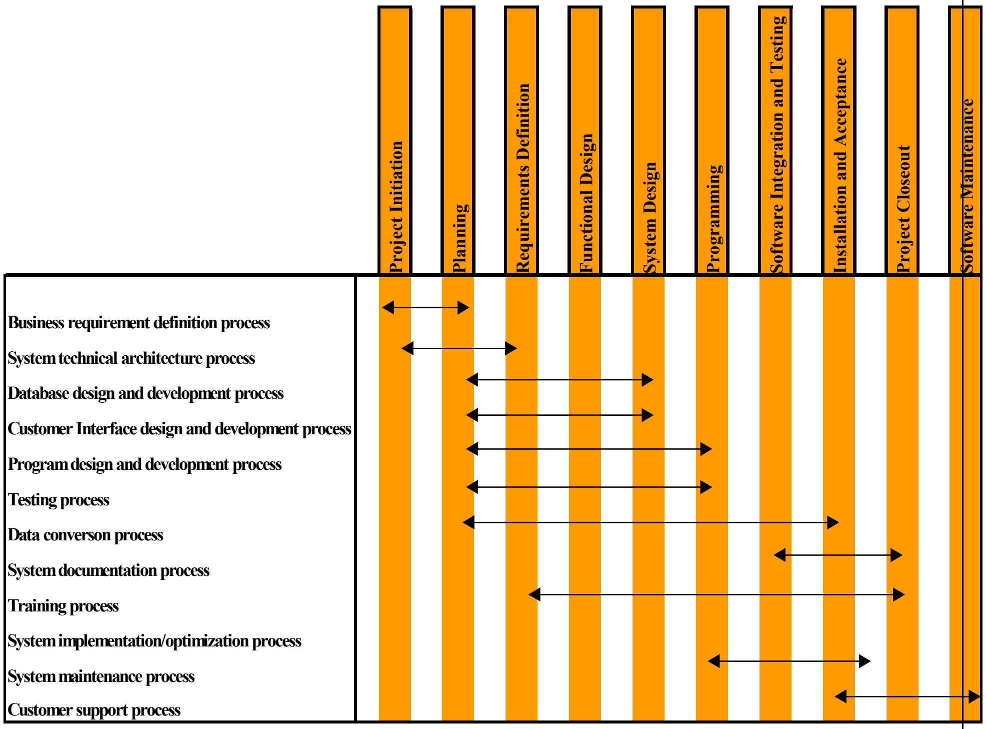
Finding Additional Information on CMM and the SEI

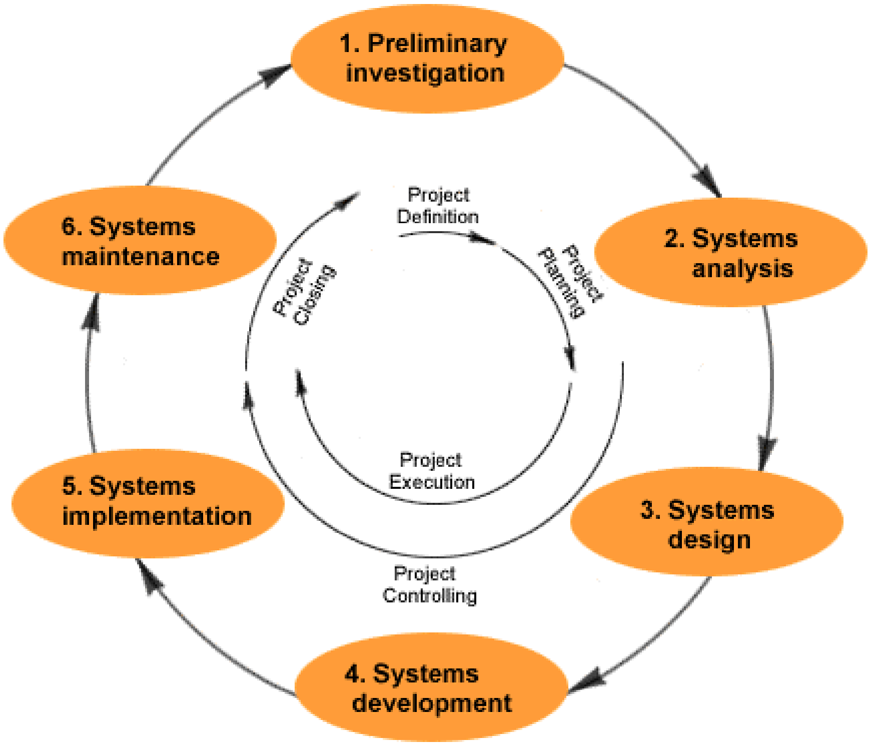
To find out more about the Capability Maturity Model and implementing its methods into your workplace, you can visit their website at http://www.sei.cmu.edu/

Notes

Large Project Guide

Large software development projects are included in the system owner's organizational long-range plans. Enterprise-wide and agency-specific projects are usually developed as large-sized projects and are likely to require a major acquisition of hardware and software. Typically, the larger the size and scope of the project, the greater the detail and coordination needed to manage the project. As risk factors and levels of effort increase, the scope of project management also increases and becomes a critical factor in the success of the project. (see Appendix L – Large Project Guide for more details)


Project Initiation
Planning
Requirements Definition
Functional Design
System Design
Programming
Software Integration and Testing
Installation and Acceptance
Project Closeout
Software Maintenance
Business requirement definition process
System technical architecture process
Database design and development process
Customer Interface design and development process
Program design and development process
Testing process
Data converson process
System documentation process
Training process
System implementation/optimization process
System maintenance process
Customer support process


1. Preliminary investigation
2. Systems analysis
3. Systems design
4. Systems development
5. Systems implementation
6. Systems maintenance
Project Definition
Project Planning
Project Execution
Project Closing
Project Controlling

This work breakdown structure is intended for projects that are 750 hours or greater. This work plan and others are available on the web site at: http://www.michigan.gov/dit.

I Project Initiation/Definition Phase

I-0	Project Initiation/Definition Activity
I-05	Project initiated
I-10	Create Project Description Statement
I-20	Create Project Concept Document
I-30	Project Concept Document Completed
I-40	Create Project Appraisal Package
I-50	Project Appraisal Package Approved
I-60	Create Project Feasibility Document
I-70	Project Feasibility Document Completed
I-80	Create Project Charter Document
I-90	Project Charter Document Approved
I-100	Steering Committee Approval
I-120	Project Management
I-130	Manage Project
I-140	Initiation Completed

P Planning

P-0	Objectives and Scope
P-1	Planning Started
P-2	Develop Critical Success Factors
P-3	Develop Project Scope Statement
P-4	Project Scope Statement Developed
P-5	Develop Project Work Plan
P-6	"Develop WBS, Assign Resources, Schedule"
P-7	Develop Organizational Breakdown Structure
P-8	Develop Cost Benefit Analyses
P-9	Project Work Plan Developed
P-10	Develop Procedural Plans
P-11	Develop Configuration Management Plan
P-12	Develop Risk Plan
P-13	Develop Quality Plan
P-14	Develop Communications Plan
P-15	Procedural Plans Developed
P-16	Requirements
P-17	Prepare Input/Output Definitions
P-18	Document Business Rules
P-19	Develop Business Reengineering Proposal

P-20	Customer Accepted Requirements
P-21	Analysis/Design
P-22	Prepare Systems Overview Documents
P-22-1	Prepare Web Design
P-23	Prepare Database Design
P-24	Prepare Screen Layout
P-25	Prepare Report Layout
P-26	Prepare Program Overview
P-27	Prepare Technical Considerations Document
P-28	Develop Final Time/Cost Estimates
P-29	Revise Project Plan and Rebaseline
P-30	Customer Accepted Design
P-31	Implementation Planning
P-32	Develop Initial Implementation Plan
P-33	Develop Initial Training Plan
P-34	Complete Initial Transition Checklist
P-35	Initial Implementation Plan Approved

E Execution

E-0	Program Instructions
E-1	Write Programming Instructions
E-2	Programming Instructions Completed
E-3	Development
E-4	Create Database
E-5	Write Programs
E-6	Create Screens
E-7	Create Reports
E-8	Conduct Unit Testing
E-9	Development Completed
E-30	System Testing
E-32	Test System
E-35	Testing Completed
E-45	Customer Acceptance Testing
E-50	Prepare User Manual
E-55	Prepare Test Environment
E-60	Assist Customer with Acceptance Testing
E-65	Code/test Corrections
E-70	Assist Customer with Regression Testing
E-75	Review Implementation Plan
E-80	Review Training Plan
E-85	Review Transition Checklist
E-89	Customer Accepted System
E-90	Implementation

E-91	Execute Implementation Plan
E-95	Execute Training Plan
E-99	Implementation Completed

C Control

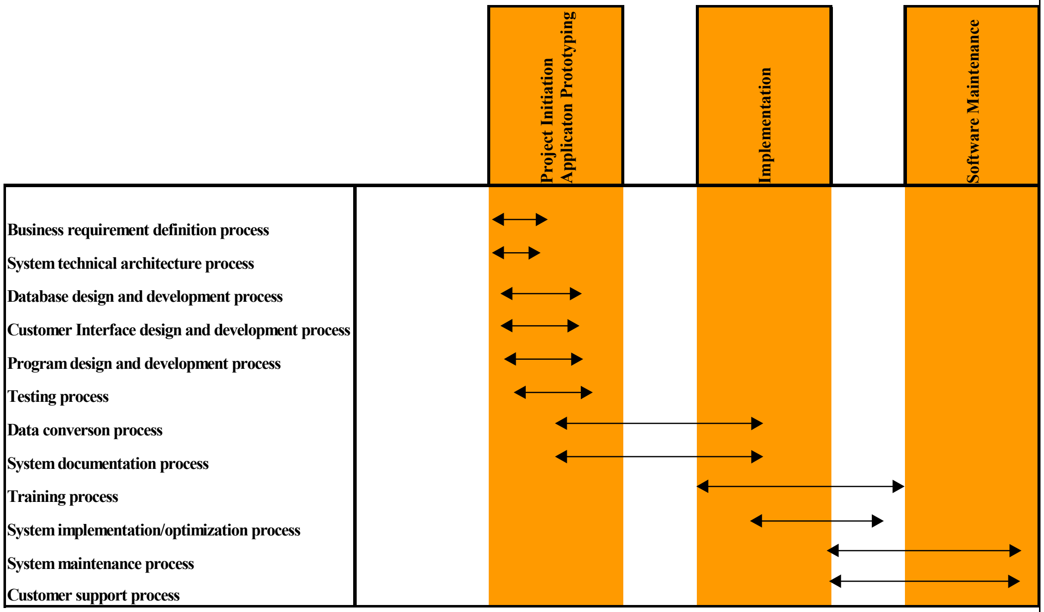
C-0	Project Control Activities
C-1	Project Control Started
C-2	Update Project Work Plan
C-3	Conduct Status Meetings
C-4	Perform Metrics and Status Reporting
C-5	Perform Change Control
C-6	Perform Scope Control
C-7	Perform Quality Control
C-8	Perform Schedule Control
C-9	Perform Cost Control
C-10	Perform Risk Control
C-11	Perform Contract Administration
C-12	Contingency Budget
C-13	Project Control Completed

CL Closeout

CL-0	Administrative Closure
CL-1	Perform Administrative Closure
CL-2	Perform Financial Closure
CL-3	Perform Financial Audit
CL-4	Archiving
CL-5	Personnel and Facilities
CL-6	Project Administratively Closed
CL-7	Post Implementation
CL-8	Post Implementation Support
CL-9	Document Lessons Learned
CL-10	Assemble Post Implementation Evaluation
CL-11	Project Completed

Small Project Guide

Small software development projects require minimal effort and use existing hardware and software. The project manager can easily manage the operational details of a small project, so formal documentation requirements are limited. A project is small when the software being developed will have limited functionality and use, meets a one-time requirement, or is developed using reusable code. (see Appendix N – Small Project Guide for more details)


Project Initiation
Applicaton Prototyping
Implementation
Software Maintenance
Business requirement definition process
System technical architecture process
Database design and development process
Customer Interface design and development process
Program design and development process
Testing process
Data converson process
System documentation process
Training process
System implementation/optimization process
System maintenance process
Customer support process